RADAR SCATTERING AND ANTENNA MODELING ON SCALABLE HIGH PERFORMANCE COMPUTERS

Tom Cwik, Cinzia Zuffada, Daniel S. Katz and Jay Parker

Jet Propulsion Laboratory

California Institute of Technology

Pasadena CA 91109

1. INTRODUCTION

The application of advanced computer architecture and software to a broad range of electromagnetic problems has allowed more accurate simulations of electrically larger and more complex components and systems than had been previously available. Design and analysis in radar scattering and antenna systems can directly and repeatedly benefit from the largest and fastest computer architectures that become available. This evolution of computational methods and their use is expected to continue unfettered into the future. The work in this chapter draws from the Parallel Applications Technology Program at the Jet Propulsion Laboratory. Initially the work in parallel computational electromagnetic at the Jet Propulsion Laboratory explored the application of early distributed memory parallel computers to existing algorithms. This work continued with the development of methods for

modeling the scattered or radiated fields from objects with penetrable or inhomogenous materials, and with geometries that required modeling at the sub-wavelength scale size. To this end finite element methods were developed, purposely crafted for use with high performance, large memory parallel computer systems. The guiding principle of this work was the development of methods that limited approximations at the theoretical, formulation stage, and implemented algorithms which allow generality of application when using high performance parallel machines. The sections that follow outline the development, implementation and results of these activities.

II. ELECTROMAGNETIC SCATTERING AND RADIATION

The problem of calculating electromagnetic fields scattered from a target, or radiated from an antenna can be solved by various means. When the object is composed of inhomogenous penetrable materials or has geometric variations on a fraction of a wavelength, finite element methods are advantageous. A finite element method calculates fields in a volumetric region in and around the object, accounting for material parameters and shape. Far fields are then found easily by integrating equivalent sources on the computational boundary.

Though the scattering and radiation problems differ greatly in application, the fundamental formulations are very similar, differing only by the location and description of the source. After a formulation of the general problem is developed, the source term is modeled appropriately and inserted

into the system of equations. For scattering problems, the source is external to the scatterer, traditionally a plane wave having a specified polarization of the electric and magnetic fields, incident on the scatterer from some direction. For antenna problems, the impressed current source exists at some specified point on the antenna. This source is appropriately modeled, typically by a known field distribution impressed on a surface, or through volumetric currents.

The remainder of this section presents the formulation for a finite element solution to the scattering and radiation problem. Initially the scattering problem is developed, with the source terms for radiation briefly outlined. The work in this section draws on [1 ,2,3] wherein additional material is presented and computational results for scattering and radiation geometries are found.

A. Formulation Of The Problem

The scatterer and surrounding space are broken into two regions: an interior part containing the scatterers and freespace region out to a defined surface, and the exterior homogeneous part (Figure 1). To efficiently model fields in the exterior region, the surface bounding the interior is prescribed to be a surface of revolution. The following formulation first outlines the interior finite element representation, then the exterior integral equation model, and finally the coupling of fields at the boundary separating the two regions.

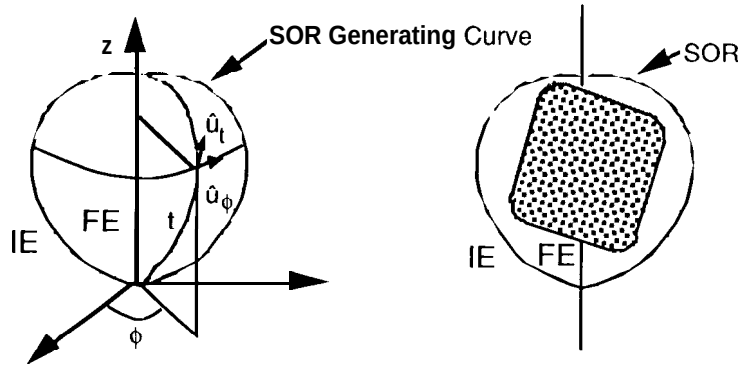


Figure **1. Geometry of computational** domain **showing interior** and **exterior regions.**

The Interior Region

In the interior region, a finite element discretization of a form of the wave equation is used to model the geometry and fields. Applying a form of Green's theorem–multiplying the wave equation by a testing function $\overline{T}$, integrating over the volume and using the divergence theorem–a weak form of the wave equation is obtained. This form includes a surface integral which provides for a boundary condition relating the field inside the selected volume to the field on the boundary, and thus provides a link to the outside field

$$\frac{\eta_0}{jk_0}\iiint_V\left[\frac{1}{\varepsilon_r}\left(\nabla\times\overline{H}\right)\bullet\left(\nabla\times\overline{T}^*\right)-k^2\mu_r\overline{H}\bullet\overline{T}^*\right]dv-\iint_{\partial V}\overline{E}\times\hat{n}\bullet\overline{T}^*ds=0\ . \qquad (1)$$

$\overline{H}$ is the magnetic field (the $\overline{H}$-equation is used in the development; the dual $\overline{E}$-equation can be similarly developed), $\overline{T}$ is a testing function, the asterisk denotes conjugation, and $\overline{E}$ x $\hat{n}$ is the tangential component of $\overline{E}$ on the surface

S (∂V). In general, ∂V represents all boundaries of the volume, including the surface of revolution and any perfect conductors. The surface integrals over the perfect conductors are identically zero since their integrand includes the tangential electric field $\bar{E} \times \hat{n}$. Equation (1) therefore represents the fields internal to and on the surface S. These fields will be modeled using a set of properly chosen finite element basis functions. In Equation (1), ε_r and μ_r are the relative permittivity and permeability respectively, and k_0 and η_0 are free-space wave number and impedance, respectively.

The Exterior Region

In the formulation of the integral equation, fictitious electric $\left(\bar{J} = \hat{n} \times \bar{H}\right)$ and magnetic $\left(\bar{M} = -\hat{n} \times \bar{E}\right)$ surface currents, equivalent to the tangential magnetic and electric fields just on the exterior of the boundary surface, are defined on the boundary. These currents produce fields in the exterior region which are the scattered fields. The sum of the scattered and the incident field results in the total field everywhere outside the boundary surface. On the boundary itself, this sum is equal to half the total field. The scattered fields are obtained from the tangential currents via an integral over the boundary using the free-space Green's function kernel. Two equations are obtained for the electric and magnetic fields on the boundary–the electric field integral equation (EFIE) and the magnetic field integral equation (MFIE), respectively. A linear combination of the two with a constant weighting factor α results in the

combined field integral equation (CFIE). The general form used in this formulation is

$$Z_M[\bar{M}/\eta_0] + Z_J[\bar{J}] = V_i \qquad \textbf{(2)}$$

where Z_M and Z_J are the integro-differential operators used in defining the CFIE, and V_i represents the incident field.

Enforcing Boundary Conditions

The previous two sections have outlined field representations for the interior and exterior regions. In the interior region, boundary conditions at any material interface, including perfect conductors, must be enforced by a proper application of the finite element basis functions. At the artificial surface of revolution separating the interior and exterior regions, boundary conditions on the continuity of tangential field components must be enforced.

Initially, four equations are written for the three unknown field quantities of interest. The first unknown is the magnetic field internal to the volume ***v***. The other two are the electric and magnetic surface currents, $\bar{J}$ and $\bar{M}$, on the boundary.

The four equations are found from the finite element equation (l), the integral equation (CFIE) relating $\bar{J}$ and M currents to the incident field (2), and a set of equations enforcing the continuity of $\bar{H}$ and $\bar{E}$ across the boundary.

Continuity of the magnetic field across the boundary is enforced in a weak sense

$$\iint_{\partial\mathcal{V}} \left(\hat{n} \times \bar{H} - \bar{J}\right) \bullet (\hat{n} \times \bar{U}^*) ds = 0 \tag{3}$$

where $\bar{U}$ is a testing function. This is an essential boundary condition and must be explicitly enforced. Continuity of the electric field across the boundary is made implicit in the finite element equation in the surface integral term $\hat{n} \times \bar{E}$, and is termed a natural boundary condition

$$\iint_{\partial\mathcal{V}} \left(\bar{E} \times \hat{n} - \bar{M}\right) \bullet \bar{T}^* ds = 0 \tag{4}$$

This equation is combined with (1) to produce

$$\frac{\eta_0}{jk_0} \int_{\mathcal{V}} \left[\frac{1}{\hat{\varepsilon}_r} \left(\nabla \times \bar{H}\right) \bullet \left(\nabla \times \bar{T}^*\right) - k^2 \mu_r \bar{H} \bullet \bar{T}^* \right] dv - \int_{\partial\mathcal{V}} \bar{M} \bullet \bar{T}^* \, ds = 0. \tag{5}$$

Equations (5), (3) and (2) constitute the system of equations representing fields in all space in and about the scatterer.

B. Why This Formulation Addresses The Problem

As outlined above, the finite element method of solving for scattered fields is chosen to capture fine-scale geometry or to model inhomogenous materials in and about the scatterer or antenna. This is accomplished through

the volume integral in (5); the material parameters everywhere within the computational volume are modeled, including any perfect conducting surfaces. The magnetic fields within the volume are then calculated in the presence of the materials without approximation, other than that of the numerical discretization and numerical solution. The mesh is ‘(pulled” out from the scatterer or antenna to encompass the volume contained within a minimal surface of revolution that surrounds the scatterer or antenna. This surface, and the integral equation representation of the fields on that surface, are used to accurately model the radiation condition for fields scattered or radiated from the object. This formulation allows an accurate implementation of the radiation boundary condition which is essential for a high fidelity solution of the fields.

Additional, the choice of a surface of revolution allows for a more numerically efficient integral equation as opposed to using a general surface. Indeed, on a surface of revolution the unknowns can be described as having a variation along the surface generator totally independent of that along the azimuthal coordinate. The azimuthal variation is expressed in terms of a Fourier series with orthogonal harmonics. As will become clear later, these features give rise to impedance matrices of order lower than those corresponding to a surface of general shape, for the same size.

Ill. FINITE ELEMENT MODELING

The above theoretical formulation is discretized into a complete finite element model, ultimately creating a large system of equations that is solved.

The following sections outline the internal finite element model and the surface integral equation discretizations.

A. Discretization of the Problem

The three equations (5), (3) and (2) are discretized using appropriate sets of basis functions. In the interior region, tetrahedral, vector edge elements (Whitney elements) are used. On the bounding surface of revolution, a set of functions with piecewise linear variation along the surface of revolution generator, and with an azimuthal Fourier modal variation are used.

The Finite Element Model

An ensemble of elements filling the interior region, excluding any perfect conducting objects, is created using a mesh generator. The elements should accurately represent the magnetic field, the geometry of the scatterer, and the bounding surface of revolution. Since the scatterer is not a body of revolution in general, the finite element mesh will extend out from the scatterer to the surface of revolution. For an accurate model of the fields, tetrahedral, vector edge elements are used to model $\overline{H}$ [4]

$$\overline{H}(r) = \sum_i h_i \overline{W}_i(r) \tag{6}$$

where

$$\overline{W}_{mn}(r) = \lambda_m(r)\nabla\lambda_n(r) - \lambda_n(r)\nabla\lambda_m(r) \qquad (6a)$$

and $\lambda(r)$ are the tetrahedral shape functions. Testing functions are also chosen to be the functions $\overline{W}(r)$.

These functions are used in the volume integral of (5). The resultant discretized volume integral is

$$\frac{\eta_0}{jk_0}\int_V \left[\frac{1}{\varepsilon_r}\left(\nabla\times\overline{H}\right)\bullet\left(\nabla\times\overline{W}^*\right) - k^2\mu_r\overline{H}\bullet\overline{W}^*\right]dv \Rightarrow \mathbf{KH} \quad (7)$$

where **K** is the assembled sparse finite element matrix, and **H is** the vector of complex, finite element basis function amplitudes.

An Efficient Exterior Integral Equation Model

To describe the surface of revolution geometry, a cylindrical coordinate system (ρ,ϕ,z) is selected for the exterior region, and orthogonal surface coordinates (t,ϕ) are used on the boundary itself; ϕ is the azimuthal angle variable and t is the contour length variable along the generating curve of the surface of revolution. In the formulation of the integral equation, the equivalent electric and magnetic surface currents ($\overline{J}$ and $\overline{M}$) are defined just on the outside of the surface through the relations

$$u\bar{E} = \bar{E}^i - L[\bar{J}] + K[\bar{M}/\eta_0] \tag{8}$$

$$u\eta_0 \bar{H} = \eta_0 \bar{H}^i - K[\bar{J}] - L[\bar{M}/\eta_0] \tag{9}$$

in which u is the Heaviside function,

$$u = \begin{cases} \text{I, for points outside } \partial V \\ \frac{1}{2}, \text{for points on } \partial V \\ \text{O, for points inside } \partial V \end{cases} \tag{9a}$$

and L and K are integro-differential operators given by

$$L[\cdots] = j\eta_0 \iint_{\partial V} \left(k_0^2[\cdots] + \nabla\nabla' \cdot [\cdots]\right) g(k_0|\bar{r} - \bar{r}'|)\, ds' \tag{10}$$

$$K[\cdots] = \eta_0 \iint_{\partial V} [\cdots] \times k_0 \nabla g(k_0|\bar{r} - \bar{r}'|)\, ds'. \tag{11}$$

In (10) and (11) g is the well known Green's function for unbounded space.

From the above derivations we can write the electric field integral equation and magnetic field integral equation, respectively, to obtain the $\bar{J}$ and $\bar{M}$ surface currents. They are

$$(\tfrac{1}{2}\eta_0\, \hat{n} \times I - K)[\bar{M} / \eta_0]\big|_{tan} + L[\bar{J}]\big|_{tan} = \hat{n} \times \bar{M}_i, \tag{12a}$$

$$(\hat{n} \times L)[\bar{M} / \eta_0] + (\tfrac{1}{2} \eta_0 I + \hat{n} \times K)[\bar{J}] = \eta_0 \bar{J}_i, \qquad (12b)$$

in which, for the sake of symmetry, the source terms (tangential components of the incident field) are given as fictitious surface currents $\bar{J}_i$ and $\bar{M}_i$. They are presented in a form that is very similar both in terms of dimensions as well as vector orientations. The symbol, /, represents the unity operator and is introduced for notational consistency.

These two integral equations are linearly combined through a weighting factor α , and the resulting sum is cast into the compact form given by (2) where the operators

$$Z_M = \{(1-\alpha)(\tfrac{1}{2} \eta_0 \hat{n} \times I - K) + \alpha (\hat{n} \times L)\}$$

$$Z_J = \{(1-\alpha)L + \alpha (\tfrac{1}{2} \eta_0 I + \hat{n} \times K)\} \qquad (14)$$

and source term

$$\bar{V}_i = (1-\alpha)\hat{n} \, x \, \mathrm{M}; + \alpha \eta_o \bar{J}_i \qquad (15)$$

are used. This formulation of the operators follows from the CICERO code development [5].

Using the method of moments, this integral equation is turned into a matrix equation. The unknown currents M and $\bar{J}$ are expanded in a finite series of basis functions $\bar{U}$ on the surface of revolution. The testing functions are selected identical to expansion functions on the surface of revolution. They are written as separable functions of t and ϕ and will have two orthogonal components along the $\hat{t}$ and $\hat{\phi}$ directions. The azimuthal function is the exponential harmonic $\exp(jn\phi)$ (Fourier harmonics). The variation along the surface of revolution generator is represented by a triangle function $T(t)$ divided by $\rho(t)$, the radial distance from the z-axis. Thus,

$$\bar{J}(\bar{M}) = \sum_{n,k} a^{t}_{n,k}(\eta_o b^{t}_{n,k})\overline{U^{t}}_{n,k} - a^{\phi}_{n,k}(\eta_o b^{\phi}_{n,k})\overline{U^{\phi}}_{n,k} \qquad in \quad \partial V \tag{16}$$

$$\bar{V}_i = \sum_{n,k}\left(c^{t}_{n,k}\,\bar{U}^{t}_{n,k} - c^{\phi}_{n,k}\,\bar{U}^{\phi}_{n,k}\right) \qquad in\ \partial V \tag{17}$$

and both expansion and testing functions are given as

$$\bar{U}^{t(\phi)}{}_{n,k} = \hat{t}(\hat{\phi})\frac{T_k(t)}{\rho(t)}e^{jn\phi} \tag{18}$$

$T_k(t)$ is a triangle function spanning the k-th annulus on the surface of revolution. Each annulus spans two segments along the generator, each referred to as a strip. Adjacent triangles overlap on one segment. These overlapping triangle functions result in approximations to $\bar{J}$ and $\bar{M}$ which are piecewise linear in f and a Fourier series in ϕ.

The original integral equation is transformed into a set of linear equations for each of the Fourier modes since the Fourier modes are orthogonal and decouple. Thus, in compact form it can be written as:

$$\sum_m m_m \langle Z_{Mm}[\bar{U}_m], \bar{U}_n \rangle + \sum_m j_m \langle Z_{Jm}[\bar{U}_m], \bar{U}_n \rangle = \sum_m v_{im} \langle \bar{U}_m, \bar{U}_n \rangle \qquad (19)$$

where m_m and j_m are the complex unknown amplitudes for each Fourier mode. This is the second equation in the system, representing fields scattered from the object.

Coupling the Two Representations

The surface integral in (5) and the first component of the integral in (3) are termed the coupling integrals, since with a convenient choice of the unknown in the first and of the testing function in the second, they are made to couple interior and exterior field representations. The surface S in these surface integrals is chosen to be that of the surface of revolution. Because the surface of revolution is discretized when using these basis functions, the issue arises of how to represent $\bar{W}$ on S. Indeed, the outer surface of the interior volume is a union of finite element facets. These facets vary, according to the order of finite element representation chosen, from planes to curved surfaces. In general, however, this surface is not identical to the surface of revolution. Similarly, the surface of revolution is obtained by revolving a generating curve around an axis, creating a surface whose cross section is circular. However,

for numerical purposes, the generator itself is not necessarily smooth, but is piecewise linear. Thus, only in the limit of fine meshing will the two surfaces coincide with each other.

The finite element function $\bar{\bar{W}}$ is evaluated approximately on the portion of surface of revolution projected from the triangular facet of the tetrahedron onto a strip. This is accomplished by an orthogonal projection of the tetrahedral facet surface onto the surface of revolution, thus introducing an error which depends on the size of the tetrahedral facet with respect to the curvature of the surface of revolution. The coupling term is given by the integral

$$C = \iint_{\partial V} \bar{U} \bullet \bar{\bar{W}}^{*} ds \tag{20}$$

where, for each integral equation basis and finite element testing function, the contributing surface is the union of the projections of a triangular boundary surface onto the proper number of surface of revolution strips (up to two in well-posed cases). Such surfaces are curved triangles, curved quadrilaterals, or curved pentagons. The evaluation of the integrals was done numerically by first inscribing the above irregular surfaces into curved rectangles and then by determining the points inside the region of interest from the knowledge of the simplex coordinates of the original finite element boundary facet and their properties at points inside the facet. These coupling integrals, as well as the discretization of the second surface integral in (3), complete the discretization of the problem.

The Complete System of Linear Equations

Having introduced the basis and testing functions for the volume as well as the surface unknowns, substitution into the complete set of equations yields

$$\begin{vmatrix} \mathbf{K} & \mathbf{C} & \mathbf{0} \\ \mathbf{C}^{\dagger} & \mathbf{0} & \mathbf{Z}_0 \\ \mathbf{0} & \mathbf{Z}_M & \mathbf{Z}_J \end{vmatrix} \begin{vmatrix} \mathbf{H} \\ \mathbf{M} \\ \mathbf{J} \end{vmatrix} = \begin{vmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{V}_i \end{vmatrix} \tag{21}$$

where

$$\begin{aligned} \mathbf{K} &= \left\langle K_p\left[\overline{W}_p\right] \cdot \overline{W}_q \right\rangle \\ \mathbf{C} &= -\eta_0 \left\langle \overline{U}_m \cdot \overline{W}_q \right\rangle \\ \mathbf{Z}_0 &= \eta_0 \left\langle \overline{U}_m \cdot \left[\hat{n} \times \overline{U}_n\right] \right\rangle \\ \mathbf{Z}_M &= \left\langle Z_{Mm}\left[\overline{U}_m\right] \cdot \overline{U}_n \right\rangle \\ \mathbf{Z}_J &= \left\langle Z_{Jm}\left[\overline{U}_m\right] \cdot \overline{U}_n \right\rangle \end{aligned} \tag{22}$$

The symbol † indicates the adjoint of a matrix. Note that both **K** and **C** are sparse, $\mathbf{Z}_0$ is tri-diagonal, and $\mathbf{Z}_M$ and $\mathbf{Z}_J$ are banded. In particular the system is complex, non-symmetric, and non-Hermitian. The sparsity of the system (21) is shown in Figure 2 for a case with only several hundred finite element unknowns. For larger, more representative cases, the number of finite element unknowns will grow into hundreds of thousands while the number of columns in C will be several hundred to several thousand.

The parallel solution to this matrix equation system is completed in two steps. Initially H in the first equation in (21) is written as H== $-\mathbf{K}^{-1}\mathbf{C}\mathbf{M}$ and substituted into the second equation resulting in

$$\begin{vmatrix} \mathbf{Z}_{\mathbf{K}} & \mathbf{Z}_{\mathbf{0}} \\ \mathbf{Z}_{\mathbf{M}} & \mathbf{Z}_{\mathbf{J}} \end{vmatrix} \begin{vmatrix} \mathrm{M} \\ \mathrm{J} \end{vmatrix} = \begin{vmatrix} o \\ \mathrm{Vi} \end{vmatrix} \tag{23}$$

where $\mathbf{Z}_{\mathbf{K}} = -\mathbf{C}^{\dagger}\mathbf{K}^{-1}\mathbf{C}$. This relatively small system is then solved directly for M and J. By solving the system in two steps, the interior solution is decoupled from the incident field Vi, allowing for efficient solutions when many excitation fields are present as in monostatic radar cross section simulations.

The relative numbers of unknowns in H and M (or J) makes the calculation of $\mathbf{K}^{-1}\mathbf{C}$ the major computational expense. This operation is the solution of a system of equations, $\mathbf{K}\mathbf{X}$ == C, where $\mathbf{C}$ is a rectangular matrix with a potentially large number of columns in the case of electrically large scatterers. The solution is accomplished by using a symmetric variant of the quasi-minimum residual iterative algorithm. The resulting overall matrix (23) is treated as being dense, and the solution of this second system is accomplished via a direct dense LU decomposition, since its size is relatively small.

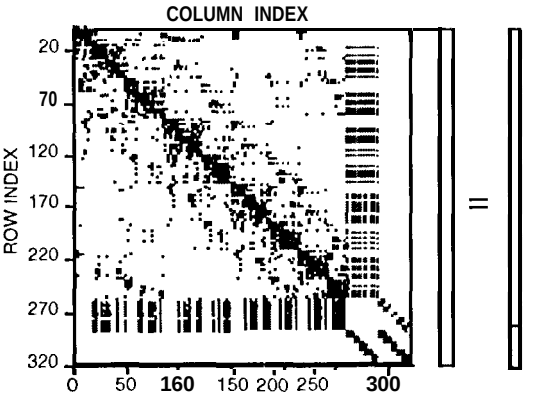


Figure 2. Scatter plot graphically showing structure of system of equations. Darkened spaces indicate non-zero matrix entries.

Modeling Source Term for Antenna Modeling

The source being modeled is an unknown field distribution on an aperture represented by a mathematical surface, one of those bounding the computational domain. It can be chosen to be the transverse cross section, or a portion thereof, of a waveguide or coaxial cable (or many of them, for arrays) feeding a radiating structure. This option is convenient because a representation of the transverse fields in terms of waveguide modal functions with unknown coefficients is known for certain geometries. Such a representation can then be used as a constraint on the finite-element solution for the field at the surface itself, as is outlined in the following.

Consider a waveguide directed along the ξ axis, transitioning into a radiating structure, and let SW (ξ = O) be the surface representing the aperture (for example, a complete waveguide cross section or an iris) chosen to

terminate the computational domain. This means that no mesh exists beyond this point looking towards the source. For $\xi < 0$ the total tangential fields can be expressed as functions of the incident mode of unit amplitude propagating along ξ and an infinite number of modes originating at the discontinuity between the waveguide and the radiating element, propagating or evanescent along the direction - ξ

$$\bar{E}_{t,w} = Z_0 e^{-\gamma_0 \xi} \bar{e}_0 + \sum_{i=1}^{\infty} R_i Z_i e^{-\gamma_i \xi} \bar{e}_i \tag{24}$$

$$\bar{H}_{t,w} = e^{-\gamma_0 \xi} \bar{h}_0 - \sum_{i=1}^{\infty} R_i e^{-\gamma_i \xi} \bar{h}_i \tag{25}$$

In (24) and (25) $\bar{e}_i, \bar{h}_i$ are respectively the TE and TM modal functions associated with the waveguide geometry, Z_i are the modal impedances and R_i are the unknown modal reflection coefficients. For all the definitions the reader is referred to [2]. From (25), since the modes are orthogonal

$$R_i = -\int_{S_0} \bar{H}_{t,w} \cdot \bar{h}_i + \delta_{0i} \tag{26}$$

is obtained, where the Kronecker delta is used.

By making use of (24) - (26) the term $\bar{E} x \hat{n}$ in the surface integral in (1) can be expressed as

$$\bar{E} \; x \; \hat{n} = Z_0 \bar{h}_0 + \sum_{i=0} Z_i R_i \bar{h}_i \tag{27}$$

Therefore, with proper substitutions, the surface integral on SW can be written as

$$\int_{S_w} 2Z_0 \bar{T}^* \cdot \bar{h}_0 ds - \sum_{i=1} Z_i \int_{S_w} \bar{T}^* \cdot \bar{h}_i ds \int_{S_w} \bar{H}_{t,w} \cdot \bar{h}_i ds \tag{28}$$

By imposing continuity between $\bar{H}_{t,w}$ and tangential $\bar{H}$ at SW expressed by the finite elements, the above expression finally reduces to

$$\int_{S_w} 2Z_0 \bar{T}^* \cdot \bar{h}_0 ds - \sum_{i=1}^{\infty} Z_i \int_{S_w} \bar{T}^* \cdot \bar{h}_i ds \int_{S_w} \bar{H} \cdot \bar{h}_i ds \tag{29}$$

it is noted that (29) introduces additional terms to the volume integral of (5), corresponding to the unknown coefficients of the finite element edges on SW in particular, (29) shows that each unknown couples to all others on SW through the modal function $\bar{h}_i$. After discretization of the equation, the following matrix problem is obtained

$$\begin{vmatrix} K' & C & O \\ C^{\dagger} & O & Z_0 \\ O & Z_M & Z_J \end{vmatrix} \begin{vmatrix} H \\ M \\ J \end{vmatrix} = \begin{vmatrix} Vi \\ 0 \\ 0 \end{vmatrix} \tag{30}$$

It is noted that the matrix K' can be viewed as mostly sparse, with the exception of a subset associated with the edges lying on SW. Normally, the source edges represent a very small number of the overall edges and the matrix is still treated as sparse. Furthermore, **Vi** is the impressed field of the waveguide fundamental mode.

B. Why Use a Scalable Massively Parallel Processor?

The result of the above formulation and discretization is the large, sparse set of equations (21). The system has a block structure as pictured in Figure 2; as noted above, the major expense involved in the solution of (21) is an intermediate calculation that results in the system (23). l-he cost of this calculation scales directly with the electrical size of the problem being examined. By using a volumetric method, tetrahedral elements must be generated to fill the volume at a density of approximately 20-30 elements per wavelength—this results in 15, 000 tetrahedral per cubic wavelength when there are 25 elements per linear wavelength. The number of columns in the C matrix has a similar scaling relationship but one that is more tied to problem geometry. It is envisioned that the number of elements needed to model problems of interest in the aerospace field (such as electrical large wing sections with coatings, or antennas mounted on or within electrically large bodies) grows to several million, and pinnacle problems envisioned today grow to several 10's of millions of elements. When the coupling matrix entries are included, the need for storage growing into 100's of Gbytes of memory is needed and sustained floating point rates into 100's of GFLOPS per second. As outlined in the next section, these sustained rates are needed for calculations involving data structures and sparse arithmetic that do not produce performance rates (as do dense matrix algorithms) that are an appreciable fraction of the machines peak rate. It is therefore necessary that the machine peak rate be 5-10 times that of the above rate to achieve useful results.

One key goal of simulation is to limit the number of physical models built and the number of experiments conducted in the iterative engineering design process. The above memory size and performance rates are needed to produce turn-around times of designs that impacts this engineering development. The scaling of larger and larger computer models with the increasing machine capacity envisioned with future parallel systems is the driving force for using massive parallel processors for these calculations. Increased fidelity on more realistic models is the goal over the next few generations of hardware that will become available.

IV. COMPUTATIONAL FORMULATION AND RESULTS

The solution of a large sparse system is the central component of the finite element simulation. (The code described in this chapter is named PHOEBUS.) Traditionally, the dependence between mesh data and the resultant sparse matrix data has been exploited in the development of mesh partitioning algorithms [6–9]. These algorithms break the physical mesh or its graph into contiguous pieces that are then read into each processor of a distributed memory machine. The mesh pieces are generated to have roughly the same number of finite elements, and to some measure, each piece has minimal surface area. Since the matrix assembly routine generates non-zero matrix entries that correspond to the direct interconnection of finite elements (elements that do not physically touch do not generate a matrix entry), the mesh partitioning algorithm attempts to create a load balance of the sparse system of

equations. Processor communication in the algorithm that solves the sparse system is meant to be limited by the ability to minimize the surface area of each mesh piece.

The algorithm for mesh partitioning typically requires less computational time than the rest of the finite element simulation, but due to the complexity of algorithms needed to create good load balance and minimal processor communication, the development of parallel partitioning codes can be quite expensive. The complexity results from the irregularity of mesh data inherent in volumetric finite element modeling. The strategy followed in this application is to exploit the availability of a global address space by using compiler constructs to efficiently decompose the matrix data among processors of the Cray T3D [10]. Because the amount of time needed to perform the matrix decomposition is a small fraction of the overall simulation time, any minor inefficiencies in using the shared memory compiler constructs are relatively unimportant. The matrix equation solution—the major time expense of the overall simulation—and the calculation of observable are accomplished using message passing algorithms. This strategy allows the use of global addressing constructs to simplify the high complexity but computationally inexpensive portion of the simulation, i.e., the parallel finite element matrix assembly from mesh data, and the use of message passing algorithms on the portions of the simulation that require high performance. The direct decomposition of the matrix entries also results in regular data structures that are exploited by efficient communication patterns in the iterative solver.

A. Constructing the Matrix Problem

In the electromagnetic scattering application considered in this chapter, the system of equations under consideration is complex-valued, symmetric and non-definite. Because the system has these properties, and because very large systems are considered (systems up to order one-million) the quasi-minimum residual iterative algorithm is used to solve the system [11]. Each row (or column) of the matrix has a number of non-zero entries, typically sixteen for the elements currently being used, and this number is constant, independent of the mesh size. The main expense of the solution algorithm is the sparse matrix-dense vector multiplication that is inherent in this as in most other Krylov subspace iterative algorithms. The matrix decomposition used in this implementation is based on row slabs of the sparse reordered system. The reordering algorithm is used to minimize the bandwidth of the sparse system. This decomposition and reordering is chosen to minimize communication of the overlapping vector pieces in the parallel matrix-vector multiplication, reduce storage of the resultant dense vector pieces on each processor, and allow for load balance in storage and computation.

Since the right-hand-side vectors in the parallel sparse matrix equation ($KX = C$) are the columns of C, these columns are distributed as required by the row distribution of K. When setting up the row slab decomposition, K is split by attempting to equalize the number of non-zeros in each processor's portion of K (composed of consecutive rows of K). The rows in a given

processors portion of K determines the rows of C that processor will contain. As an example, if the total number of non-zeros in K is HZ, a loop over the rows of K will be executed, counting the number of non-zeros of K in the rows examined. When this number becomes approximately nz/P (where P is the number of processors that will be used by the matrix equation solver), the set of rows of K for a given processor has been determined, as has the set of rows of C.

The reordering is chosen to minimize and equalize the bandwidth of each row over the system [12]. Because the amount of data communicated in the matrix-vector multiplication will depend upon the equalization of the row bandwidth, different reordering algorithms have been examined. The generalized reverse Cuthill-McKee algorithm (in both the SPARSPAK [12] and the Gibbs-Poole-Stockmeyer [13] versions) produces an ordering that minimizes system bandwidth, and equalizes the bandwidth over each row of the matrix. Matrices resulting from objects that were long and thin, as well as those resulting from spherical objects have been examined. The nested dissection ordering in [9] could produce a smaller profile of the reordered matrix, but equalization of the row bandwidth was not accomplished; row bandwidths even approaching the matrix order were found in a few rows of the matrix.

The matrix decomposition code, termed P_ SLICE, consists of a number of subroutines. Initially, the potentially large mesh files are read (READ). Then the connectivity structure of the sparse matrix is generated and reordered

(CONNECT), followed the generation of the complex-valued entries of K (FEM), building the connectivity structure and filling the **C** matrix (COUPLING). Finally the individual files containing the row slabs of K and the row slabs of **C** must be written to disk (WRITE). For each processor that will be used in the matrix equation solver, one file containing the appropriate parts of both the K and **C** matrices is written.

Cray Research Adaptive FORTRAN (CRAFT) is used for the matrix decomposition stage of the simulation. All large arrays are declared using *CDIR$* directives to be shared in either a block manner or a cyclic manner for the leading dimension, with non-leading dimension distributed degenerately. Using a block distribution of a matrix of size 256 on 4 processors leads to the first 64 elements residing on processor O, the next 64 elements on processor 1, etc. A cyclic distribution would lead to processor O having elements (1, 5, 9, . ..). processor 1 having elements (2, 6, 10, . ..). etc. A two dimensional array with a degenerate distribution of the second dimension leads to all elements of the array having a given index in the first dimension being on the same processor, regardless of the index in the second dimension. For example, a two dimensional array of size (256,1 O) distributed degenerately over the second dimension will have elements $((i,1),(i,2),\ldots.(i,10))$ all located on the same processor. Which processor this will be is dependent on the value of i, and the method of distribution over the first dimension.

Routines which could be easily parallelized by CRAFT directives were FEM and part of COUPLING. The directive *CDIR$ DO SHARED* was added to

the parallelizable loops to automatically distribute the work over all the processors. Other routines that could be executed in parallel with a combination of CRAFT and message passing included the READ and WRITE routines. The remaining routines (CONNECT, and a second part of COUPLING) are basically sequential routines, where only one processor is doing the majority of the work, while using data spread across many (usually all) processors.

Two files are read in the READ routine, one containing finite element data, and the other containing integral equation data. The finite element file is at least an order of magnitude larger than the integral equation file, and is read by 4 processors. By using these 4 processors, the time of the READ routine is reduced roughly by a factor of 3 as compared to reading the file with 1 processor. Further reduction in this time may be possible; however, this factor of 3 is currently sufficient. In the WRITE algorithm, data is assembled on each processing element and written to disk. On the T3D, it is faster to assemble a local array and write out that data than to write out a distributed array directly, since as the number of processors increases, more writes of smaller amounts of data are being performed, and disk and network contention develops. Scaling beyond this point quickly leads to diminishing returns from each processor.

Figures 3 and 4 show the performance of P_SLICE over varying numbers of processors for two different problems. The number of edges is the number of finite element unknowns in the problem. It may be observed that for

the routines that have been parallelized, doubling the number of processors reduces the amount of time by a factor of approximately two, For routines that are sequential, where only one processor is doing the work using the other processors' data, the time goes up very slightly as the number of processors for the overall code are increased. This is due strictly to communication latency. As the number of processors increases, the percentage of array elements which are not local increases, and the time to load or store these elements is longer than the time to load or store local elements. The I/O time should have roughly the same behavior, but for practical tests the I/O time is more dependent on the 1/0 load of the other T3D processors and the load on the front-end YMP that is between the T3D and the disks than the number of T3D processors being used in P_SLICE. It is clear that the routines that benefit most from the parallel implementation on the T3D are COUPLING and WRITE.

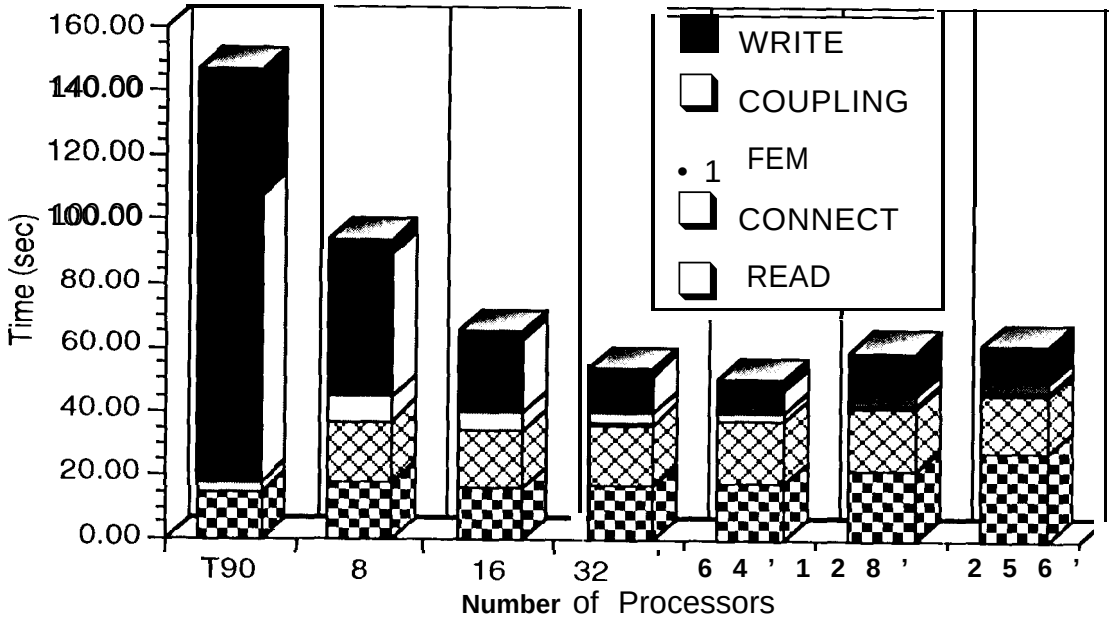


Figure 3. Computation time and scaling for a relatively small simulation (dielectric cylinder with 43,791 edges, radius = 1 cm, height = 10 cm, permittivity = 4.0 at 2.5 GHz). First column shows time for single processor T90. Times on T90 for CONNECT and FEM have been combined.

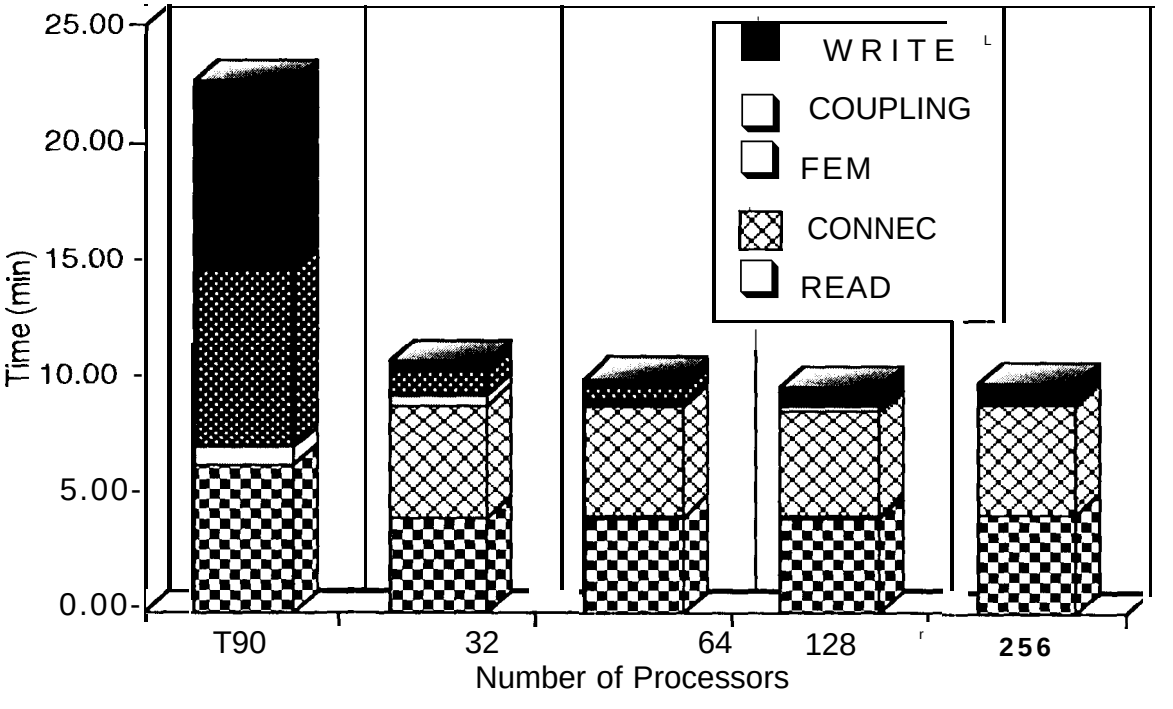


Figure 4. Computation time and scaling for a relatively arge simulation (dielectric cylinder with 579,993 edges, radius = cm, height = 10 cm, permittivity = 4.0 at 2.5 GHz). First column shows time for single processor T90. Times on T90 for CONNECT and FEM have been combined.

B. Beginning the Matrix Solution

As outlined above, the partitioned system of equations is solved in two steps, namely P_SOLVE and P_FIELD. Initially the quasi-minimum residual algorithm [11] is used to solve the sparse system of equations $\mathbf{KX} = \mathbf{C}$, resulting in the reduced sub-matrix $\mathbf{Z}_k$. The parallel quasi-minimum residual solver developed for this application operates on matrix data decomposed by row slabs in P_SLICE after reordering (Figure 5 shows matrix structure before and after reordering). The machine is logically considered to be a linear array of processors, with each slab of data residing in one of the processors. $\mathbf{C}$ and X are also decomposed by row slabs, corresponding to the row partition of the matrix. Central components of the quasi-minimum residual algorithm that are affected by the use of a distributed memory machine are the parallel sparse matrix--dense vector multiplication, and dot products and norm calculations that need vector data distributed over the machine. The dominant component is the matrix-vector multiplication, accounting for approximately 80% of the time required to run P_ SOLVE.

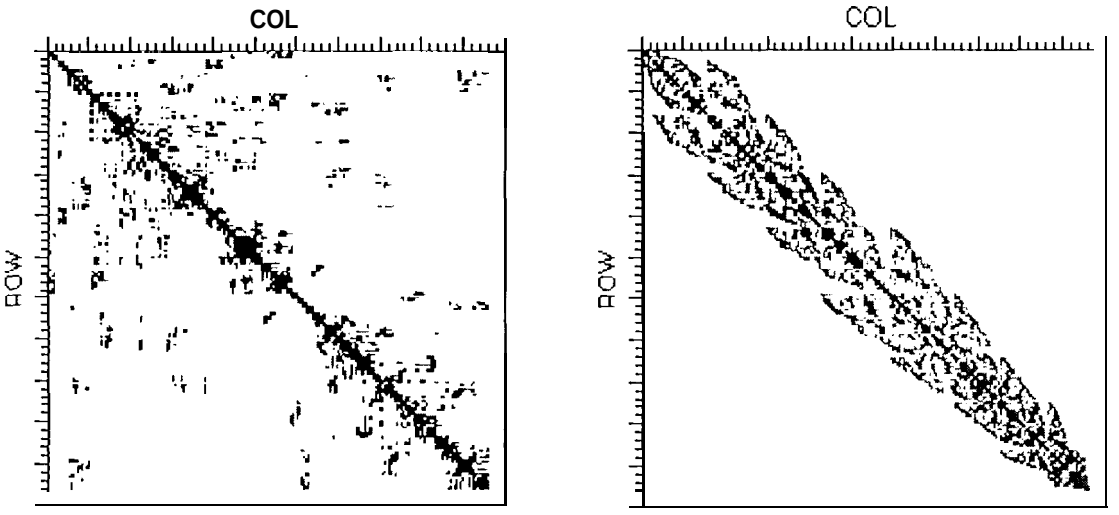


Figure 5. Original matrix structure (left) and after reordering (right).

Filled spots indicate non-zero entries of matrix.

A parallel library of the needed level-one BIAS routines was developed using CRAY T3D *shmem_put* and *shmem_get* message passing. The routines required by the quasi-minimum residual algorithm are CDOTU and SCNRM2, and the parallel implementation of these was trivial, consisting of a local BLAS call to calculate each processor's contribution to the result, and a call to a global sum routine to calculate the final result.

Parallel Sparse-Matrix Dense-Vector Multiplication Formulation

I-he parallel sparse matrix-dense vector multiplication involves multiplying the K matrix that is distributed across the processors in row slabs, each containing a roughly equal number of non-zero elements, and a dense vector x, that is also distributed over the processors, to form a product vector y, distributed as is x (Figure 6). Since the K matrix has been reordered for

minimum bandwidth, the minimum and maximum column indices of the slab are known, If the piece of the dense vector x local to this processor has indices within this extent of column indices, the multiplication may be done locally and the resultant vector y will be purely local. In general, the local row indices of the dense vector x do not contain the range of column indices; therefore a communication step is required to obtain the portions of the vector x required by the column indices of the K matrix. This communication step only requires data from a few processors to the left and right. The exact number of processors communicating data is dependent on the row bandwidth of the local piece of K, and the number of processors being used. In the simulations considered, the number of processors communicating data is typically one or two in each direction on scaled problems.

This communication could be performed using either *shmem_get* or *shmem_put*. These are one-way communication calls where the processor from whose memory the data is being gathered or to whose memory the data is being stored, respectively, is not interrupted by the communication. The *shmem_get* formulation is more intuitive and simpler to program, but the communication bandwidth of the *shmem_put* routine on the T3D is substantially higher than the communication bandwidth of the *shmem_get* routine. For this reason, the *shmem_put* formulation is used. This formulation requires the cache to be flushed to maintain cache coherency, but the resulting performance of the matrix-vector multiplication is still 15% higher than the performance obtained using the *shmem_get* formulation.

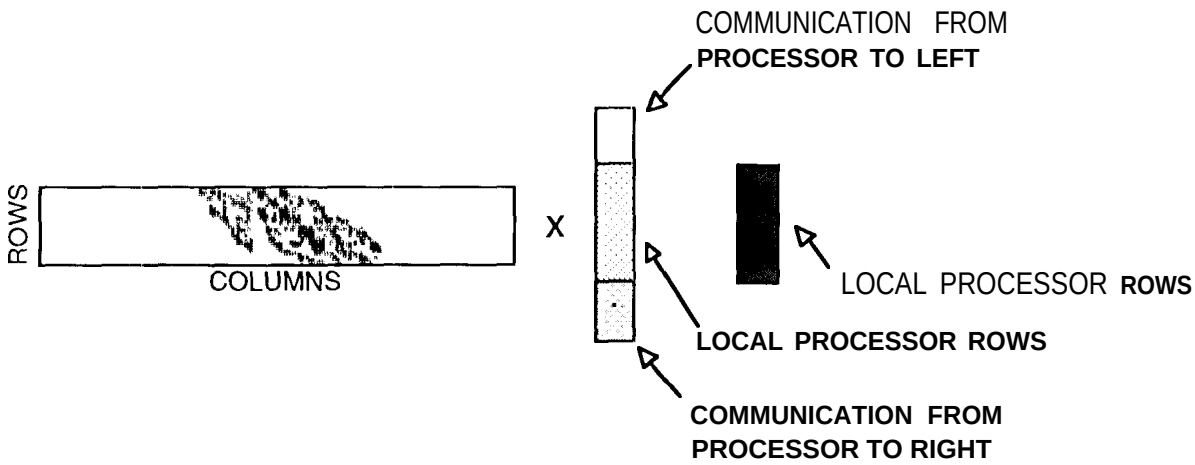


Figure 6. Local sparse matrix-dense vector multiplication graphically displayed.

As described previously, the K matrix is stored in row slabs using row-compressed storage. As K is symmetric, this is equivalent to a column slab decomposition using column-compressed storage. K may be used in either way in the matrix-vector multiplication. In this step, a non-zero in column i requires $\mathbf{x}(i)$ to be obtained, and a non-zero in row j will produce a partial result for $\mathbf{y}(j)$. This implies that K stored in column slabs will require only communication of portions of y non-local to the processor after the local portion of the multiplication, and similarly, K stored in row slabs will require communication only to gather x before the local portion of the multiplication. Since similar amounts of communication are required using either storage scheme, the scheme that minimizes the time spent in local work has been chosen for implementation. This is the row slab decomposition of **K**, because

the row-compressed storage scheme better reuses the T3D processor's local cache, and therefore has better overall performance.

Parallel Sparse-Matrix Dense-Vector Multiplication Performance

The goal of the combination reordering-partitioning strategy discussed above is to minimize as well as equalize communication in P_SOLVE, while retaining memory load balance. The partitioning chosen clearly succeeds in evenly dividing the data among the processors; Figures 7 and 8 show the relative communication time of the processors.

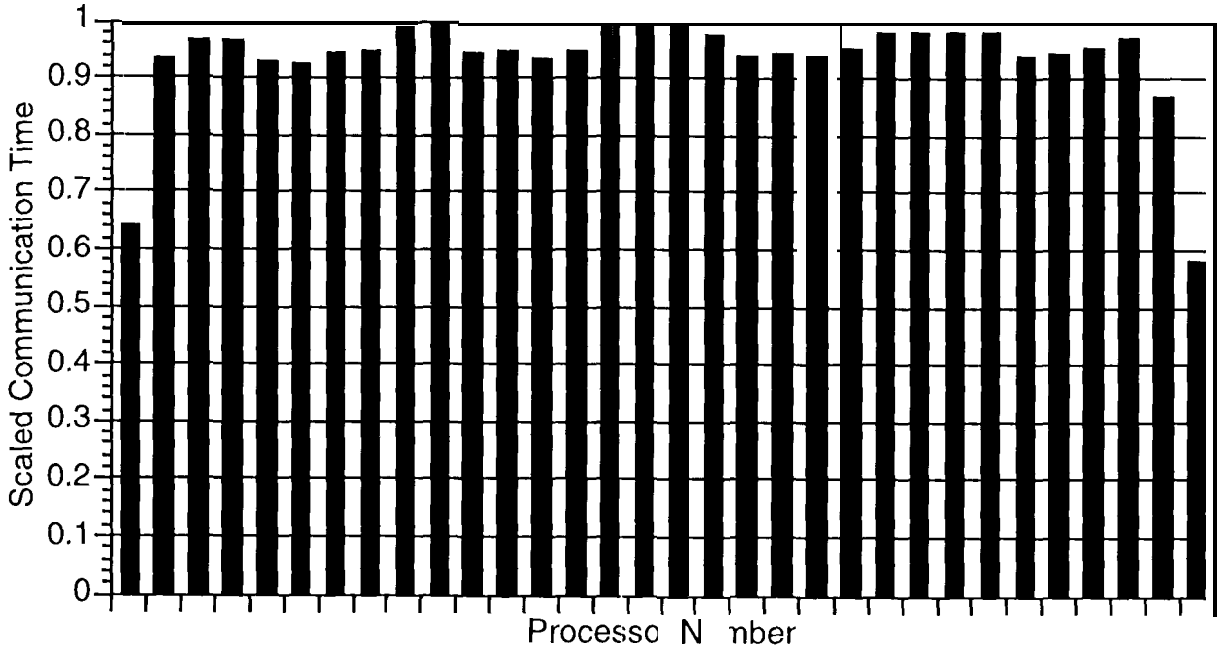


Figure 7. Graph of communication load balance for parallel matrix vector multiplication, 271,158 edge dielectric cylinder, 32 processors.

Figure 7 shows results representative of the majority of the cases that have been run. All processors, excepting those on the ends of the linear processor array, have a relatively similar amount of communication, and since the communication is synchronized, all processors will require as much time as the one that uses the most time. Only the two end processors will be idle very long at the barrier. For this case, all processors except the first and last have to communicate with two other processors, one to the left and one to the right.

Figure 8 shows the other possible class of results, shared by a minority of cases that have been run. Again, the two end processors are using less

time for communication than the majority of processors. However, in this example, a small subset of the processors are using more time in communication than the average processor. All the processors except those in this subset have to wait a substantial amount of time at the barrier, and the speed per processor of this run is lower than that of the first example. Again in this example, all processors but the first and last have to communicate with at least two other processors, one to the left and one to the right, but here, the processors in the subset that are spending more communication time are communicating with possibly two processors in either direction. The issue in these few cases is that the decomposition of the K matrix was performed entirely based on storage load balance, with the assumption that the reordering would equalize the row bandwidth and create communication load balance. This assumption is generally valid, as shown in Figure 7, though not always, as shown in Figure 8.

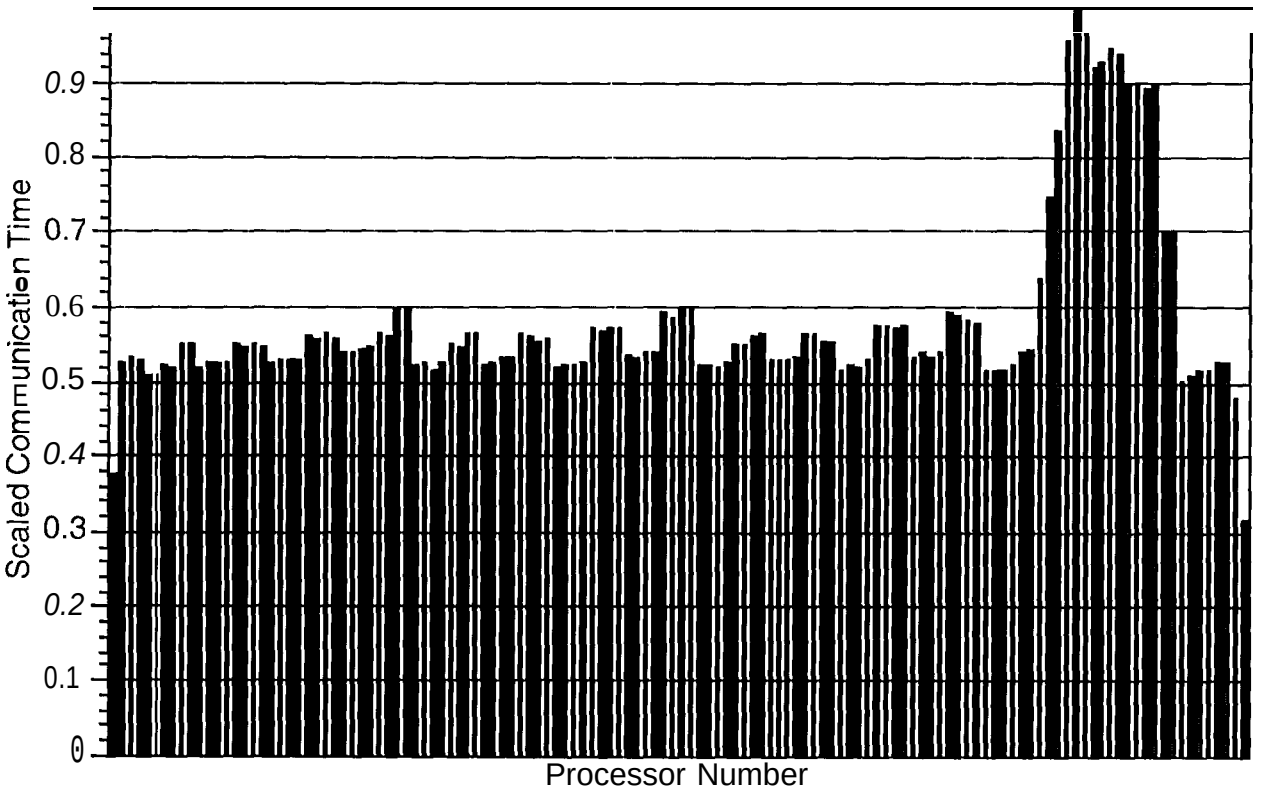


Figure 8. Graph of communication load balance for parallel matrix vector multiplication, 579,993 edge cylinder, 128 processors.

Another factor in the performance of the parallel matrix-vector multiplication is the percentage of communication. This is mainly related to the number of processors to the left and right that each processor must communicate, and as discussed above, the maximum number that any processor must communicate with. It is clear that running a fixed size problem on an increasing number of processors will generate a growing amount of communication. The amount of communication is a function of how finely the

K matrix is decomposed, since its maximum row bandwidth after reordering is not a function of the number of processors used in the decomposition. If the maximum row bandwidth is m and each processor in a given decomposition has approximately m rows of K, then most processors will require one processor in each direction for communication. If the number of processors used for the distribution of K is doubled, each processor will have approximately $m/2$ rows of K. Since the row bandwidth doesn't change, each processor will now require two processors in each direction for communication. But since the number of floating point operations required hasn't changed, the communication percentage should roughly double. This can be seen in Figure 9, which shows communication percentage versus number of processors, for four problem sizes.

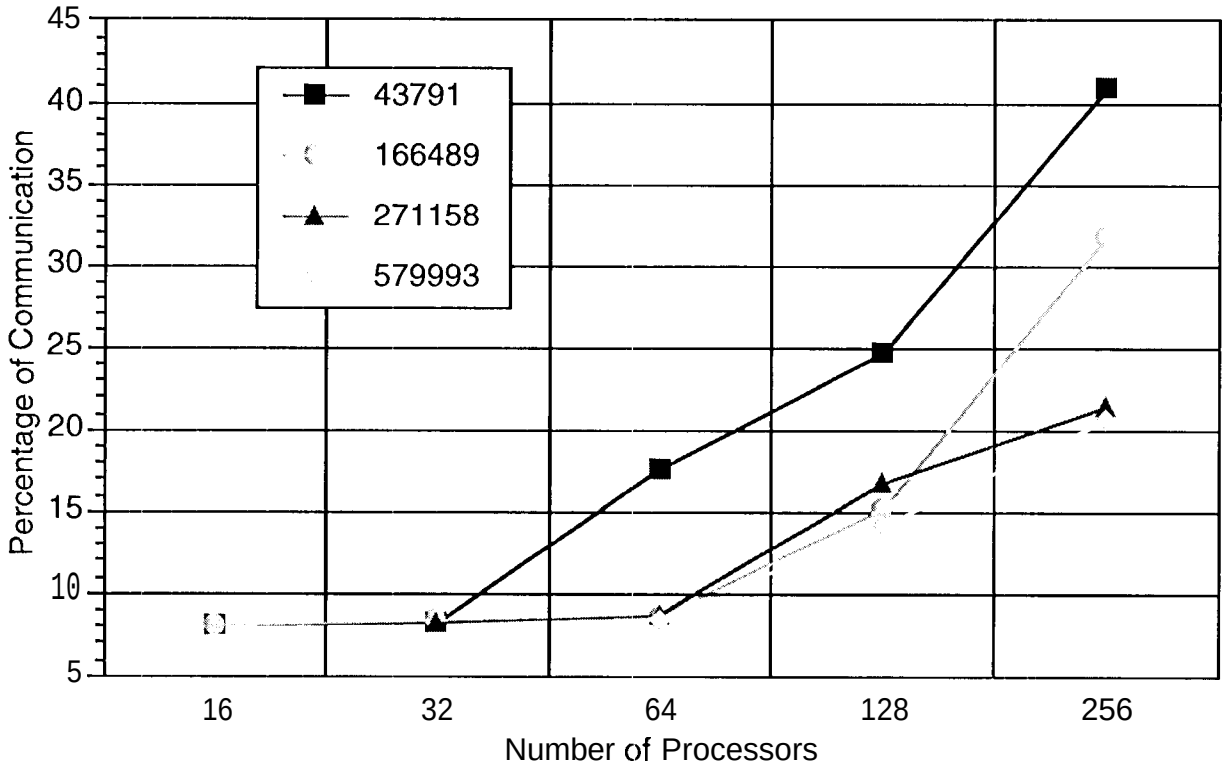


Figure 9. Percentage of communication versus number of processors for parallel matrix vector multiplication, for four different size (number of edges) meshes of dielectric cylinder.

Figure 10 shows the local rate of operations/second for the parallel matrix vector multiplication. It is measured after communication has been completed. It can be seen that the performance of this operation is roughly constant, and is not easily identifiable as a function of problem size or number of processors. To a limited extent, a problem which involved more data on each processor will run slightly faster than would a problem with less data on each processor, but as Figure 10 demonstrates, this isn't necessarily true.

The storage of the data and how it fits in the T3D's cache is more important than the amount of data, and this forces the local performance rate not to be a simple function of problem size per processor.

Shown in Figure 11 are plots of time to convergence on different numbers of processors for five different problems. The number of unknowns in the finite element mesh and the number of columns of $\mathbf{C}$ are indicated on the plots. The quasi-minimum residual algorithm was stopped when the normalized residual was reduced three orders of magnitude for each column of $\mathbf{C}$. With an initial guess being the zero vector, this results in a normalized residual of 0.1 %, a value that is sufficient for this scattering problem. Given a fixed communication percentage and a fixed rate for local work, doubling the number of processors for a given problem would halve the total solution time. The curves in Figure 11 do not drop linearly at this rate because these assumptions are not met, as shown by Figures 9 and 10. The decreased amount of work per processor causes the curves to level off as the number of processors increases.

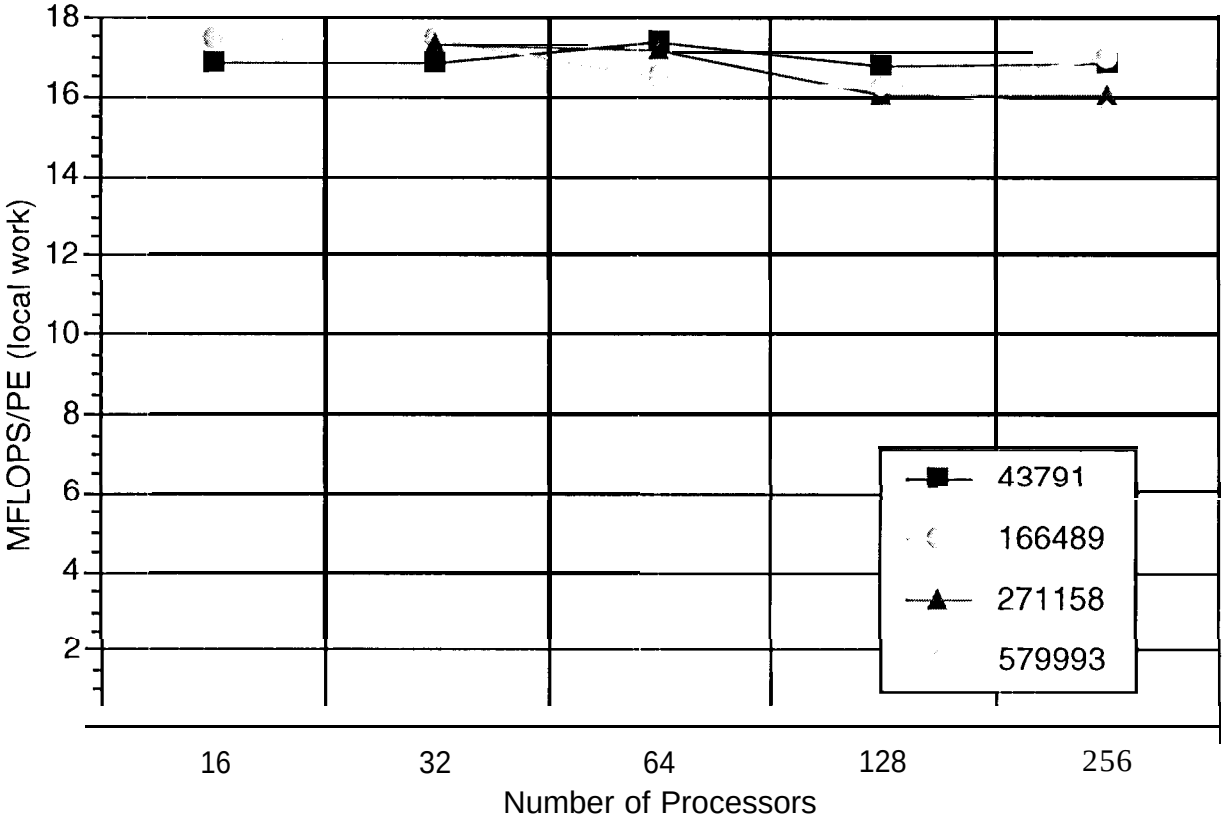


Figure 10. Local operation rate versus nurnbero fprocessors for parallel matrix vector multiplication, for four different size (number of edges) meshes of the dielectric cylinder.

Additional Work in P_SOLVE

After each column of $\mathbf{K}^{-1}\mathbf{C}$ is computed using the quasi-minimum residual algorithm, it must be multiplied by $\mathbf{C}^{\dagger}$ to obtain the equivalent column of $\mathbf{Z}_{\kappa}$. Each of these multiplies requires a global communication, since $\mathbf{C}$ is distributed over the T3D by row slabs. To reduce the number of global communications, after a number of columns of $\mathbf{K}^{-1}\mathbf{C}$ are computed, these are multiplied by C^{+}, and the columns of $\mathbf{Z}_{\kappa}$ obtained are written out sequentially to

disk. The original quasi-minimum residual algorithm solved a single solution vector at a time. A pseudo-block (multiple right-hand-side) quasi-minimum residual variant was written, which performs each quasi-minimum residual iteration on some number of columns of C simultaneously. As the residual of each column of $\mathbf{K}^{-1}\mathbf{C}$ converges below the threshold, that column is no longer used in the quasi-minimum residual algorithm. This variant performs the same number of floating point operations as the single right-hand-side quasi-minimum residual algorithm, but the K matrix is required to be loaded from memory much less often. This leads to a time savings of 10–15% in P_SOLVE.

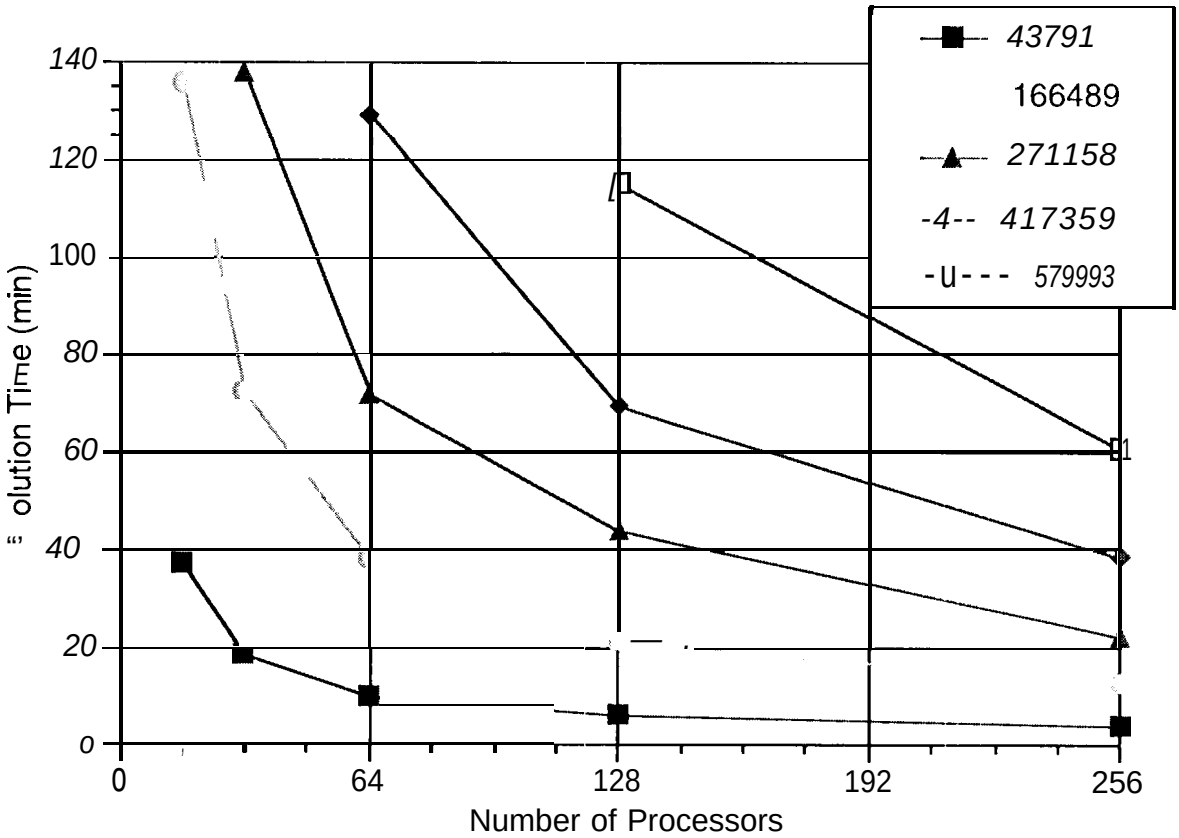


Figure 11. Time of convergence for five different problems. The time shown is the total execution time for the solver on different numbers of processors. The C matrix has 116 columns in each case.

C. Completing the Solution of the Matrix Problem

The final code of the simulation, P_FIELD, completes the matrix calculation shown in Equation (23) and computes observable quantities (radar cross section, near fields, etc.) After the $\mathbf{Z}_M$, $\mathbf{Z}_J$ and $\mathbf{Z}_0$ sub-matrices and $\mathbf{V}_i$ vector(s) are computed, and the sub-matrix $\mathbf{Z}_K$ (formed by P_SOLVE) is read in from disk, a parallel dense matrix LU decomposition algorithm is used to solve

the reduced system [14]. Since this system is much smaller than the larger sparse system solved above, the $\mathbf{Z}$ matrices may be distributed on a smaller set of processors, chosen to optimize the solve time. The time needed to solve this system compared to the sparse system is a small fraction, typically less than 1%.

The radar cross section is found from the mesh surface equivalent currents $\bar{M}$ and $\bar{J}$. This calculation—an integral over the surface—is easily parallelized on the processors executing P_FIELD. If the radar cross section for more than one excitation vector is needed (monostatic), a block of solution vectors are found, and a block of radar cross sections calculated.

D. The Three Stages of the Application

Shown in Figure 12 is the comparison of time requirements of the three stages of the simulation, for four different problem sizes. The problem simulated corresponds to the dielectric cylinder outlined in previous results. As is clearly shown, the dominant component of the simulation is P__SOLVE—the iterative solution of the sparse system. The matrix decomposition stage (P_SLICE) is relatively small, while the observable calculation stage (P_FIELD) is a minor fraction of the total time. This last stage can grow if a large number of field calculations are required, but it will typically remain a small fraction of the matrix solution time.

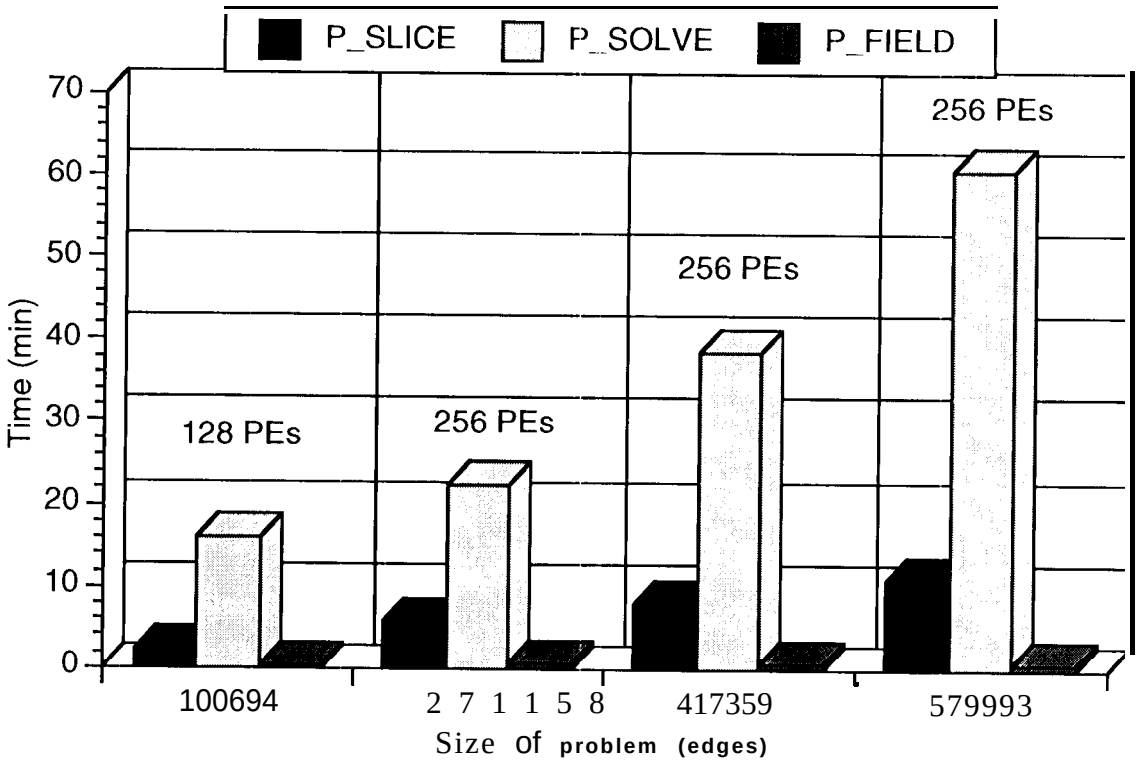


Figure 12. Comparison of time requirements for three stages of simulation for four different sizes of cylinder problem.

V. RESULTS FOR RADAR SCATTERING AND ANTENNA MODELING

Scattering results for various geometries can be found in [3]. This section presents results for scattering from anisotropic materials, and for radiation from patch antennas.

A. Anistropic Scattering

Anisotropic materials find application in electromagnetic scattering for coatings applied to targets, or in applications where specific magenetic materials are present. While there is little in the software implementation that

is unique to the combination of anisotropic materials and parallel computing, high-performance computing enables simulations of scattering and antenna problems that involve anisotropic materials and also have electrical sizes in the range of interesting problems. Thus it is interesting and practical to ensure that a high-performance parallel system performing electromagnetic scattering and radiation problems also handles anisotropic materials accurately.

As outlined in Section 1, the PHOEBUS software solves Maxwell's equations within the volume represented by the finite element mesh by the weak-form volumetric integral in (l). This formulation enforces strict tangential continuity of the primary modeled field at material boundaries, while also weakly enforcing normal continuity of the flux (in an average sense over the facets). This combination of conditions is well suited to anisotropic materials. Since we are solving frequency-domain systems, we are restricted to linear, memory less, but general anisotropic materials. Chiral and other bianisotropic materials are not supported.

The cases of an anisotropic principal-axis dielectric material surrounded by air, a radial-oriented anisotropic sphere, and an idealized gyrotropic material considered as a cavity-mode problem have been implemented and numerically verified. This latter case establishes the applicability of the finite element method used in PHOEBUS to gyrotropic materials, but was performed using separate software to solve the associated eigensystem.

Several questions of suitability were considered and answered by these test cases. Does the element-by-element weak-form integral result in stable

solutions to anisotropic problems without introducing vector parasite error? In particular, are the solutions continuous within materials, and do they display the correct discontinuities at the material boundaries? Do solutions within principal axis dielectrics display the correct divergence characteristics? This was particularly in doubt because it is well known that low-order edge elements imply basis functions that are entirely divergence-free within each element volume, while fields within principal axis dielectric volumes must be permitted to diverge; if we are primarily modelling E, then $\nabla \cdot \varepsilon E = 0$ but $\nabla \cdot E \neq 0$.

For gyrotropic materials, we have tested the accuracy of the representation for finding resonant frequencies and modes in a cavity problem. This allows systematic high-order testing of the modeling physics and demonstrates in particular the behavior with respect to spurious modes. Such modes, corresponding to extrapolated behavior at the zero-frequency limit, appear to be at the root of vector parasite problems in some alternative implementations that use node-based basis functions.

Principal axis anisotropy is incorporated in the weak-form finite element equation by direct substitution of the tensors for ε and/or μ. These tensors are fully specified by the three principal components, for example $\varepsilon_1, \varepsilon_2$ and ε_3, plus a rotation matrix **R** that specifies the material principal axes as unit vectors in the global Cartesian coordinate system. Thus

$$\varepsilon = \mathbf{R} \begin{bmatrix} \varepsilon_1 & 0 & 0 \\ 0 & \varepsilon_2 & 0 \\ 0 & 0 & \varepsilon_3 \end{bmatrix} \mathbf{R}^T \tag{31}$$

and

$$\mu^{-1} = \mathbf{R}\begin{bmatrix} \mu_1^{-1} & 0 & 0 \\ 0 & \mu_2^{-1} & 0 \\ 0 & 0 & \mu_3^{-1} \end{bmatrix}\mathbf{R}^{\mathrm{T}} \tag{32}$$

A variation is spherical anisotropy, for which every finite element volume in a sphere is assigned the same uniaxial material for which $\varepsilon_2 = \varepsilon_3,, \mu_2 = \mu_3$ and **R is** constructed separately for each element so that the first principal axis is aligned with the radius vector (the vector from the sphere center to the element centroid). This forms a good approximation to a continuous spherically anisotropic material. For this case the specification for a material is complete when $\varepsilon_1, \varepsilon_2$, μ_1, μ_2 and the coordinates of the sphere center are specified.

For gyrotropic materials, we have neglected loss and dispersion. Thus we have used μ of the form

$$\mu = \begin{vmatrix} \mu_{11}' & \chi_{12} & 0 \\ -\chi_{12} & \mu_{22}' & 0 \\ 0 & 0 & \mu_{33} \end{vmatrix}, \tag{33}$$

where $\mu'_{11} = \mu_{11} + \chi_{11}$, and $\mu'_{22} = \mu_{22} + \chi_{11}$; μ denotes the unbiased permeability (usually $\mu_{11} = \mu_{22} = \mu_{33} = \mu_0$), while x denotes the susceptibility tensor due to magnetization.

The first anisotropic test case consists of a thin dielectric slab with differing dielectic along one principal axis (Figure 13). The principal axis

relative dielectric constant *(4.0)* and propagation direction thickness ($\lambda/4$) were chosen for these test cases because of the interesting property of the corresponding infinite transverse thickness slab: polarization may be converted from linear to circular. As the linearly polarized wave from the left reaches the slab, it is conceptually split into two components, one along each of the principal axes y' and z'. The y' component sees a quarter wavelength of free space, while the z' component sees a half wavelength of dielectric constant *4.0.* Each component has perfect transmission (no reflection); but the z' wave exits 90° ahead of the y' component. Thus incident linear polarization results in emitted circular polarization. With a 1x1 wavelength slab, this property is only partially emulated, yet clearly shows the effects of the material anisotropy. A comparison is made of the fields along the propagation axis with respect to a solution by finite difference simulation on a uniform mesh of parallelpipeds. The results for the two field polarizations show excellent agreement (Figure 14)

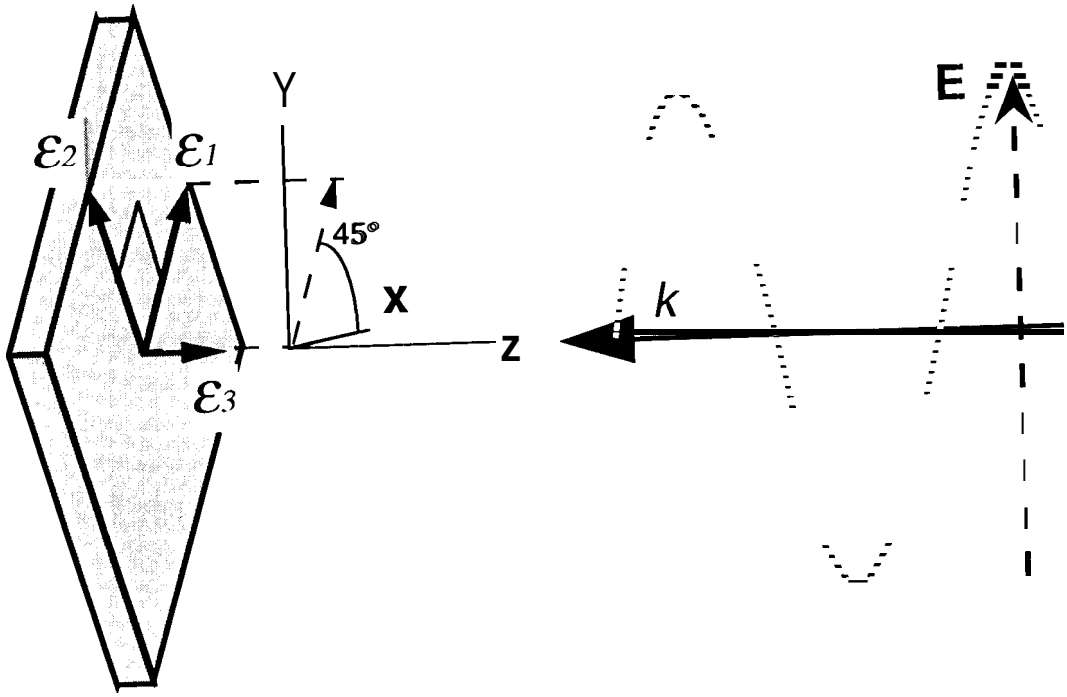


Figure 13. Geometry of anisotropic slab test cases, showing incident field, computational domain, and oblique view of slab.

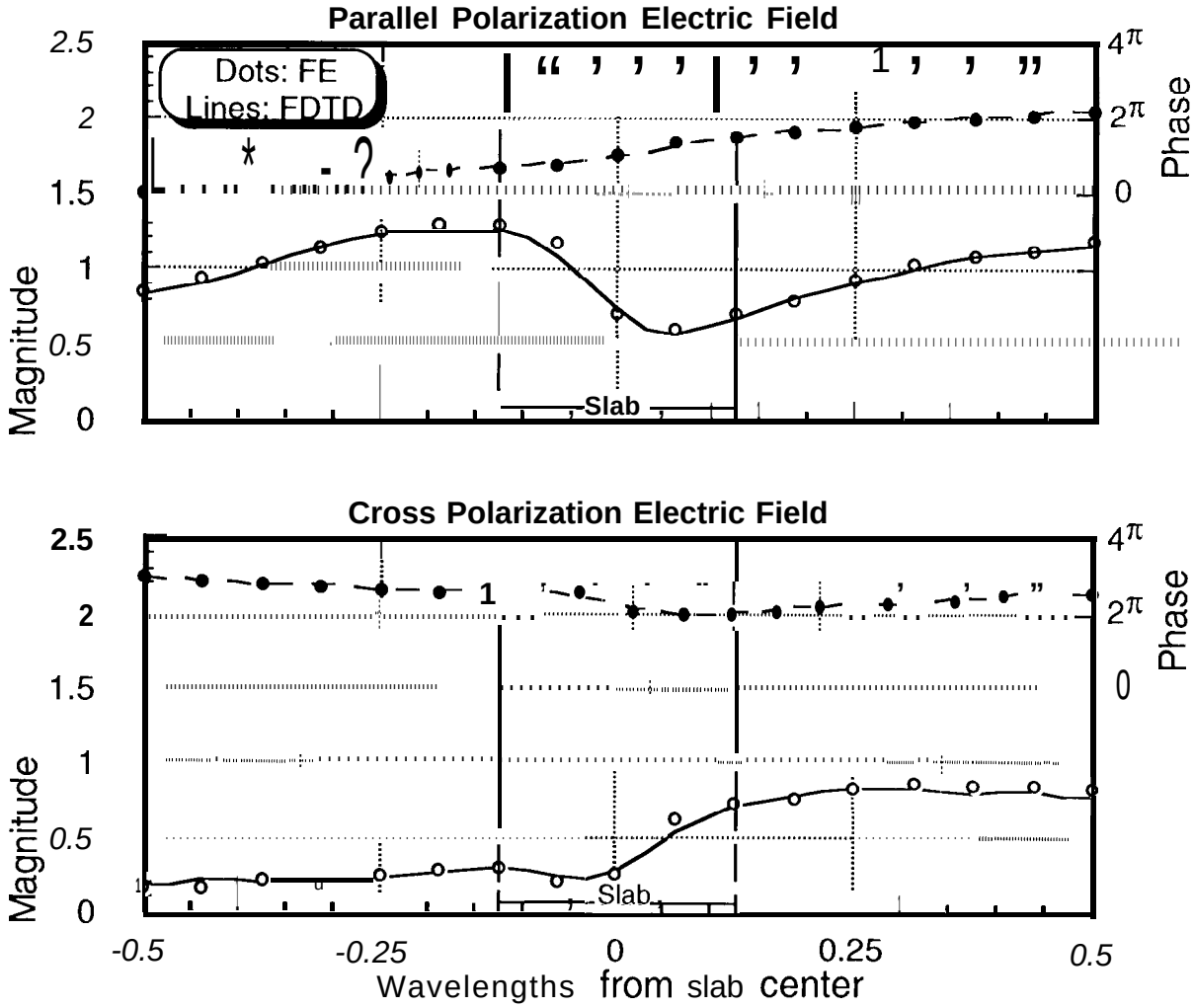
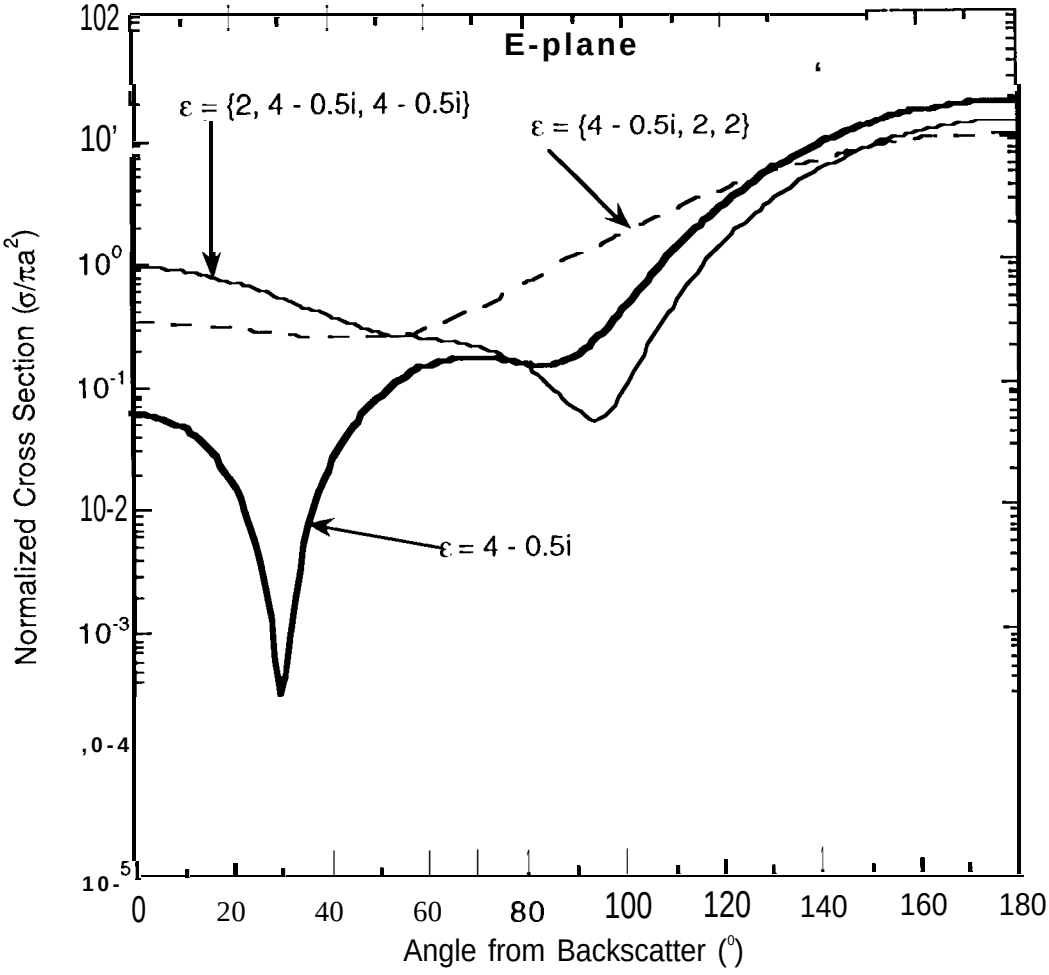


Figure 14. Electric field along propagation direction through center of slab of Figure 13. Top: magnitude and phase of total field component with polarization parallel to the incident electric field vector. Bottom: same, with cross-polarization component of total electric field.

The comparisons made to date indicate that the standard mesh density rules (8-16 elements per wavelength in each medium) is adequate for

anisotropic substances. Concerns for modeling the physically non-zero divergence in anisotropic objects by using edge elements have proved groundless: the elements support divergence as generalized functions at the mesh facets. That is, the total volume divergence over a cluster of finite element domains (i.e., the surface flux by Gauss' theorem) need not be zero, even though it is point-wise zero over each element interior. The paradox is resolved by recognizing that the divergence is supported at the facets between each adjoining pair of elements, in a generalized function sense. Essentially the normal component of the field is a step-function at the facet, hence the divergence is carried by a delta function at that point. Furthermore, the accuracy of the field treatment at dielectric interfaces is shown to be adequate and automatic. This edge-element conformability is shown to work for anisotropic-to-isotropic dielectric interfaces, as well as isotropic-to-isotropic dielectric interfaces.

A second anisotropic dielectric test case is a radially anisotropic sphere. Figure 15 shows the bistatic cross-section for several values of ε_1 and ε_2: 1) a lossy isotropic sphere with $\varepsilon_1 = \varepsilon_2 = 4 - 0.5i$, 2) $\varepsilon_1 = 4 - 0.5i$, $\varepsilon_2 = 2$ (radially lossy), and 3) $\varepsilon_1 = 2$, $\varepsilon_2 = 4 - 0.5i$ (tangentially lossy). All displayed cross-sections show reasonable agreement to those of Taylor [15, Figures 7 and 8].


E-plane
ε = {2, 4 - 0.5i, 4 - 0.5i}
ε = {4 - 0.5i, 2, 2}
ε = 4 - 0.5i
Normalized Cross Section (σ/πa²)
Angle from Backscatter (°)
102
10'
10⁰
10⁻¹
10-2
10⁻³
,0-4
10-5
0
20
40
60
80
100
120
140
160
180

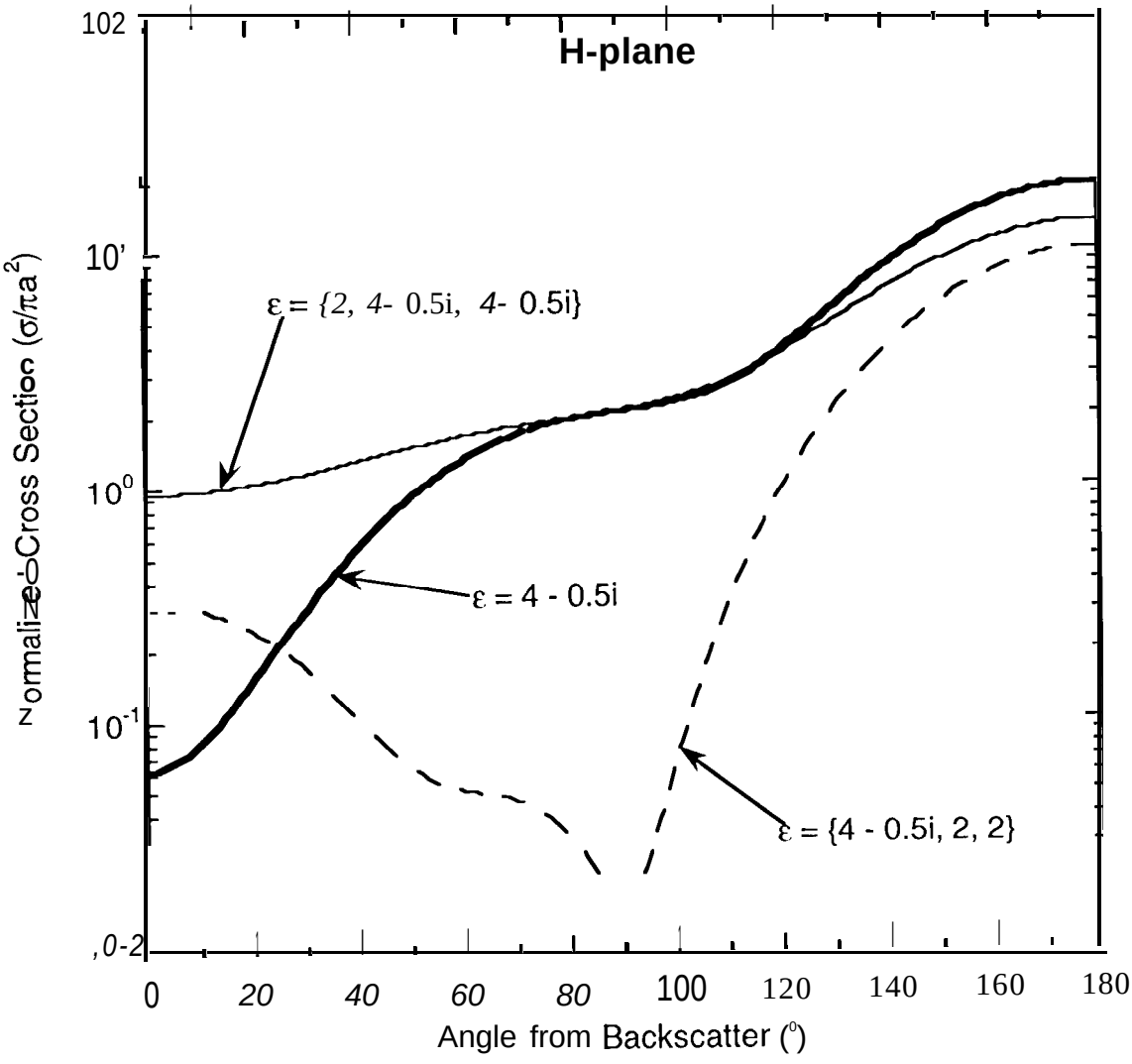


Figure 15. Bistatic cross sections for anisotropic spheres with radial, tangential relative dielectric values as shown.

The gyrotropic cavity problem consists of a height to radius ratio of 2, PEC cylinder filled with idealized gyrotropic material. The gyrotropic material

constants are $\mu_{11}' = 1.0$, $\mu_{22}' = 1.0$, $\chi_{12} = 0.1i$. This problem has an analytic solution, and has also been solved as a 2-D body of revolution finite element problem [16]. The degrees of freedom corresponding to electric fields tangent to PEC facets are eliminated from the matrix storage data structures. The finite element matrices for the eigenvalue problem corresponding to the cavity modes are **R** (resulting from the curl-curl term) and S (resulting from the direct field overlap integrals), with coefficient (eigenvalue) k^2. The resulting system, $\mathbf{R} - k^2 \mathbf{S} = \mathbf{O}$, represents a modified eigenvalue problem, which was solved by EISPACK subroutines.

We computed eigensolutions to our 3-D finite element matrices (Figure 16). The conducting ferrite-filled cylinder displays eigenvalues for modes in close agreement with those predicted analytically and found by the BOR technique from [16]. The eigensolver also finds 190 k = O modes (for the case with 190 interior finite element vertices), representing the space of functions **E =** grad(f) (for any arbitrary scalar function *f*) that satisfy the curl-curl equation for k=0. The numerical eigenvalues imply $k < 10^{-2}$ for all of these modes, demonstrating superb separation between these mathematical modes and the physical modes that have nonzero k.

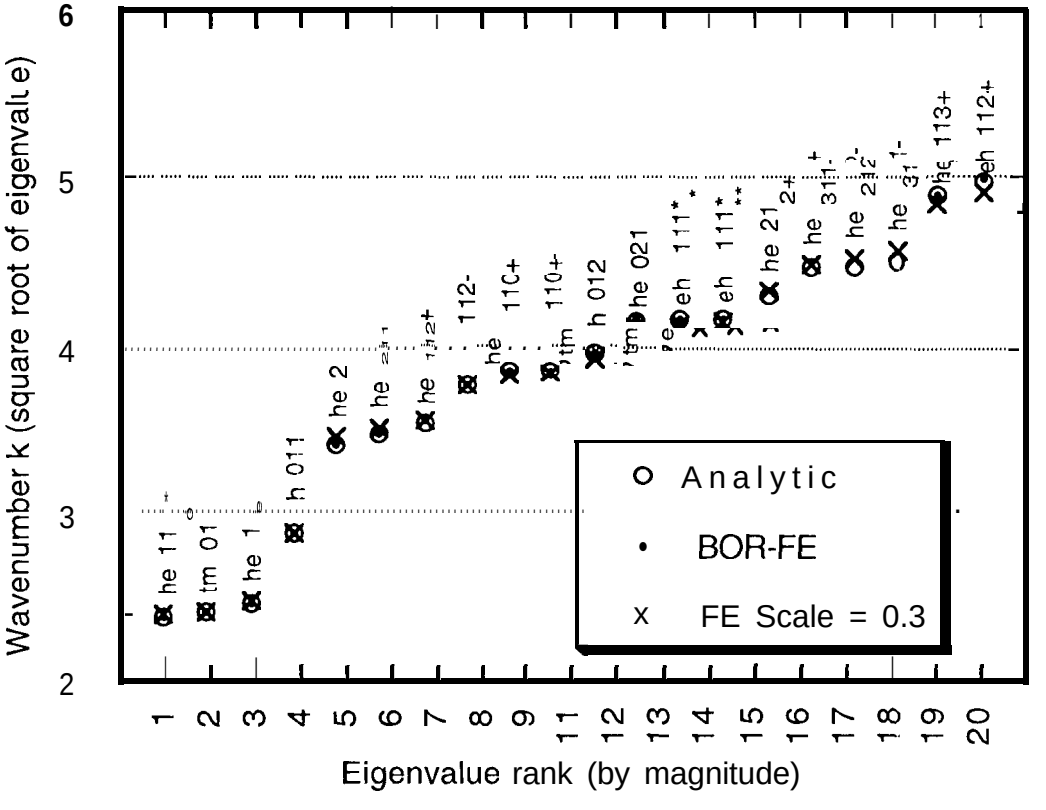


Figure 16. Ferrite cavity geometry and comparison of resonant frequencies computed analytically, by BOR finite element code and by 3-D gyrotropic prototype code

Additional validation was obtained by examining interpolated plots and animations of the interior fields on cross-sections of the ferrite-filled cylinder. These were not directly compared with analytic or computational solutions, but demonstrate whether the mode-structure agrees with that predicted by the analytic solution, for example in number and type of field nodes, and the splitting of the circular polarized modes (in a ferrite, circular modes of opposite handedness have different eigenvalues, in contrast to the degenerate modes in empty space). The first 14 modes were examined, and found to agree with the predicted mode structures and polarizations.

The quality of the results for the eigensystem indicate we may use gyrotropic materials in PHOEBUS with high confidence. Spurious modes are well behaved, and therefore vector parasitic errors should be absent.

B. Patch Antennas—Modeling Conformal Antennas With PHOEBE

Antenna modeling is accomplished using a variant of the PHOEBUS software named PHOEBE. As noted above, the major modification is that the source is now internal to the mesh which results in the linear system (30). The solution of this linear system is performed similarly to that of the scattering problem outlined above.

To simulate antennas mounted conformably on curved platforms, an array of four patches laying on top of the lateral surface of a metal cylinder and backed by a cavity, as illustrated in Figure 17 was modeled. Note that the metal patches and the cavity have the same curvature as the cylinder, thus providing perfect match of the two structures, without protrusion. Similarly, the back of the cavity is curved so that the thickness is maintained constant at 0.25 cm. The dimension of the rectangular patches and their placement with respect to the cavity is illustrated in Figure 17c. The radius of the cylinder is 10 cm and its height is 50 cm. The cavity is placed symmetrically about the middle of the platform.

A coaxial cable feeds the right-bottom patch only, while the others act as isolated parasitic elements. The higher order mode (2,0) was excited, occurring near the frequency of 4.6 GHz. A detail on how the coaxial cable attaches to the patch and the cavity is presented in cross section Figure 17d. PHOEBE requires a piece of cable to be modeled with a fine-elements mesh, up to a truncation surface, transverse to the waveguide axis, where the mesh is terminated by imposing that the modal waveguide representation pertinent to the feed geometry be consistent with the finite elements solution. In this specific case a cable length of 1 cm protruding out from the back of the cavity was included in the finite element model. The actual size of the coaxial cable is not critical; in fact one can always adjust the permittivity value of the insulator to achieve the desired characteristic impedance. In meshing the coaxial cable region, we find it useful to avoid very small values for the inner and outer radii,

and minimize the amount of mesh needed in this region. Because the cable transitions into a much larger structure – the volume of the patch – the mesh generator, in trying to obtain a transition between different edge lengths, tends to generate an unnecessarily large number of elements in the transition region.

The cylindrical platform is imagined to be surrounded by air, and we mesh a uniform layer of thickness 0.8 cm enclosing the structure, as illustrated in Figure 17e. This choice allows for a two-element thick mesh, which is desirable for accuracy. Additionally, we mesh the cavity and the coaxial cable stub. At the frequency of interest the electrical size of the cylinder is about 8 λ x 3.3 λ, With a nominal edge length specified to be $1/20^{th}$ of the wavelength in both air (outer layer) and dielectric (cavity + feed), the number of finite elements in this mesh is 260,000, corresponding to 300,000 edges.

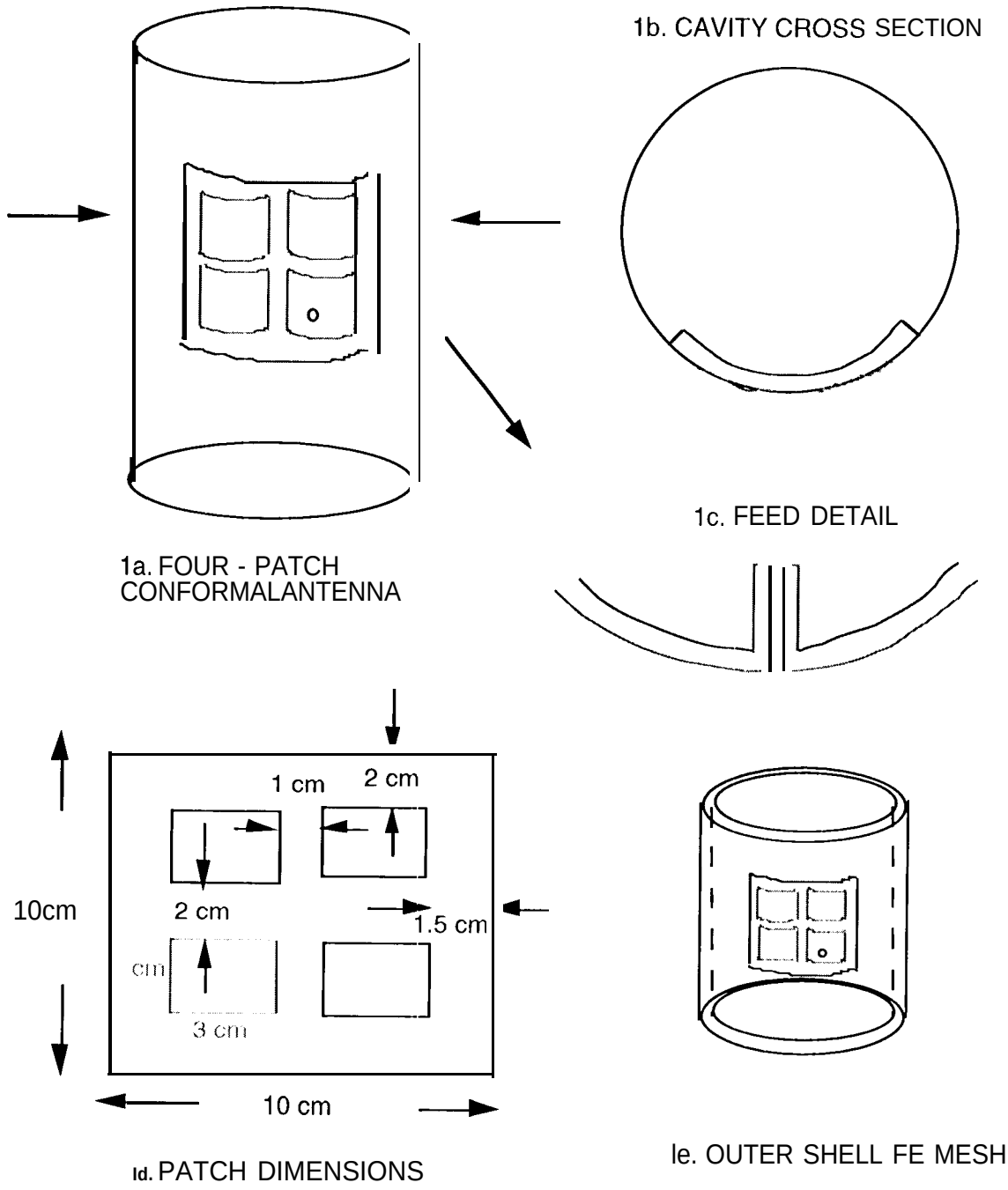


Figure 17. Geometry of conformal patch antenna,

To determine the number of Fourier modes necessary to model the radiated fields accurately, one needs to investigate the expected behavior of

the fields in and around the four patches. Noting that this is not simply a scattering problem, it is expected that some of the smallest physical dimensions present in the patch geometry, such as the azimuthal distance of 1 cm between the patches, will play a role in establishing patterns of interference varying with this scale. Based on the electrical size of the cylinder alone, at least 11 Fourier modes are required, that is to say all the modes with index between -11 and +11. In reality several more are needed, based on the antenna geometry, and we included up to 20 in the region around the cavity, tapering off as we moved along the generator away from this active region. The current coefficients for **M** and **J** constituting the solution of the matrix problem are illustrated in Figure 18, where the abscissa represents the right hand side (column of the C matrix) associated with one particular triangle (t or ϕ polarization) and Fourier series component. One can clearly see that the contribution to the solution of modes up to +/-15 is significant. Correspondingly, the radiation pattern (shown in Figure 19) is consistent with that of a patch antenna, excited in the mode (2,0). The slight asymmetry in the main lobes of the E-field pattern is attributed to the parasitic effect. Additionally, the calculated input impedance is plotted in Figure 20, and the tangential magnetic fields (phi component) calculated from the solution coefficients on the outer (truncating) lateral surface is shown in Figure 21. It is noted that the field pattern is consistent with the patch mode and that the parasitic effect is rather pronounced for the lower left patch but it is rather small

for the two upper patches, which is to be expected for the chosen excitation mode.

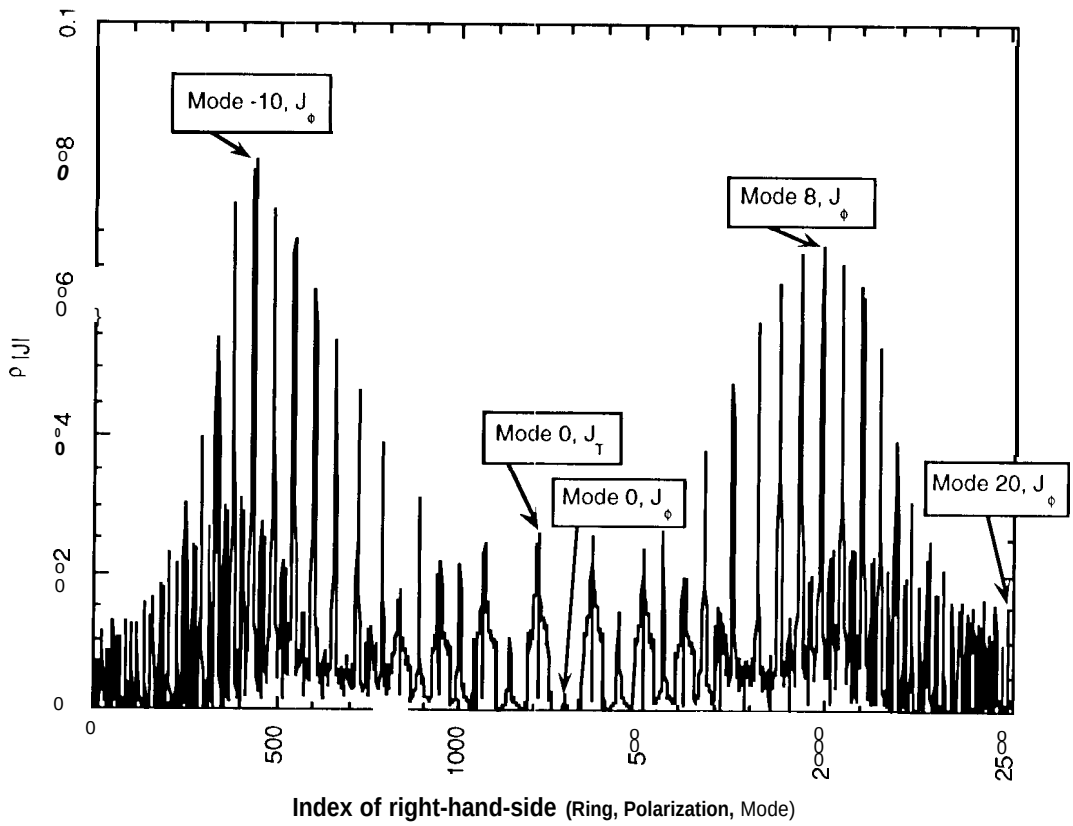


Figure 18. Amplitude of modal coefficient corresponding to index of right-hand-side of C matrix.

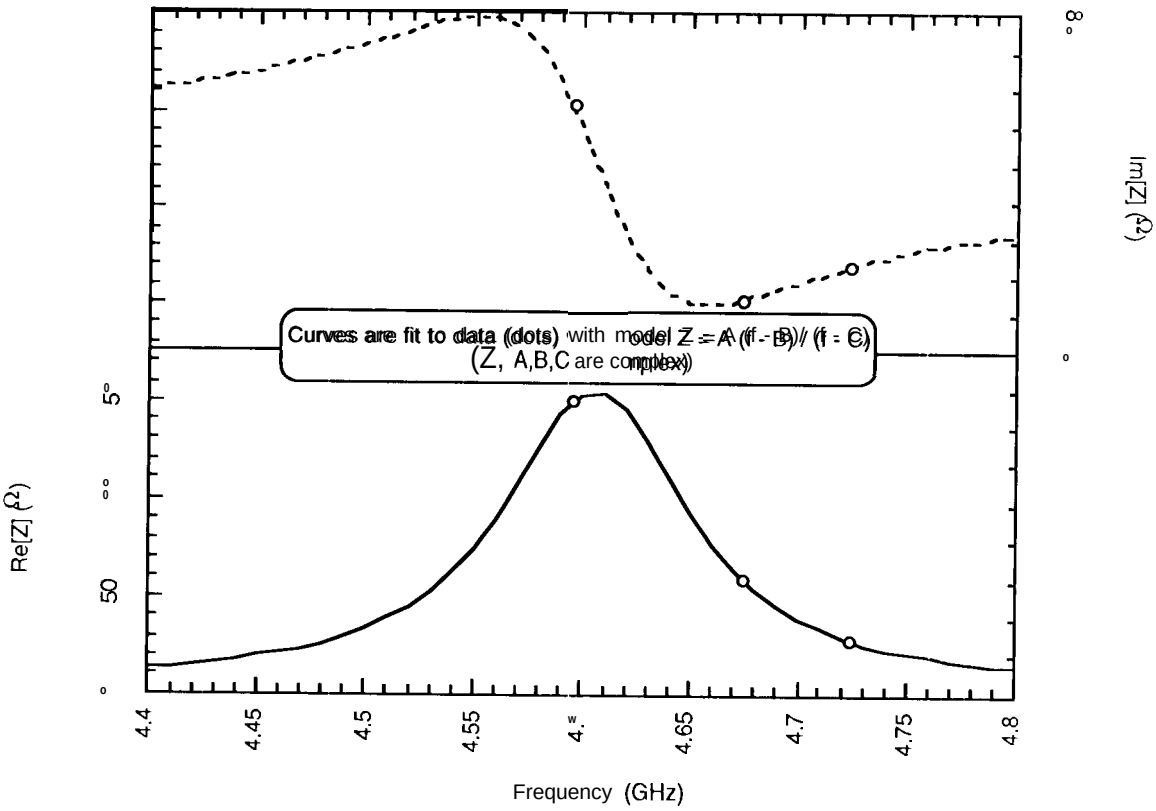


Figure 19. Real and imaginary parts of input impedance for conformal patch antenna.

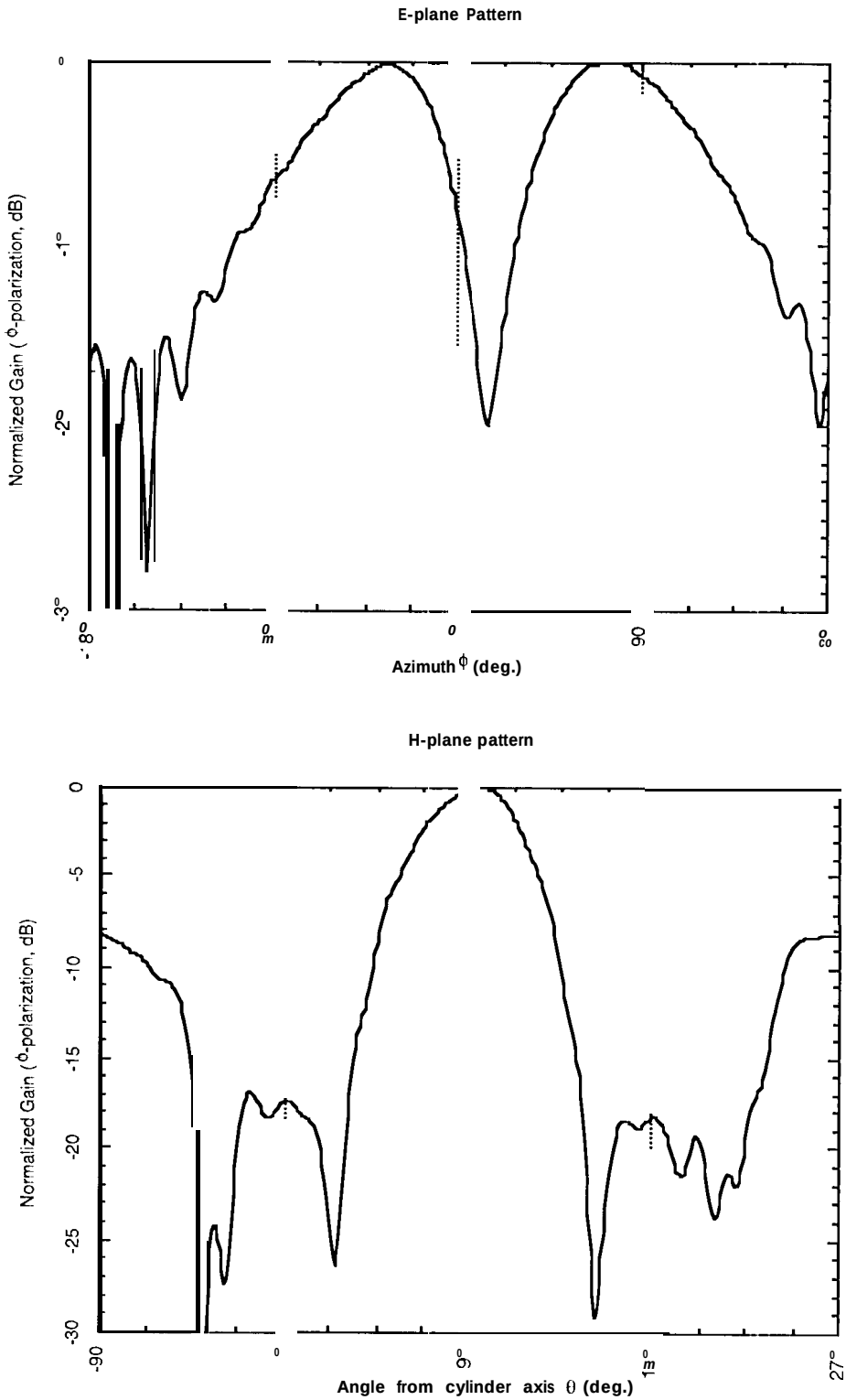


Figure 20. Eand Hplaneradiation patterns ofconformal patch antenna.

With our choice of triangles and modes along the generator, the resulting C matrix has about 2500 columns, as shown in Figure 18. We ran this problem on 128 processors of the Cray T3D and it took about 9.5 hours to complete. Most of this time, 8.3 hours, was taken up by the QMR solver on the 2,500 right hand sides of the C matrix. The solve stage was completed in multiple code runs, employing the restart feature of our QMR implementation. By comparison, filling the 5,000 x 5,000 dense matrix took about 40 minutes. A communication percentage time of slightly less than 5% was observed.

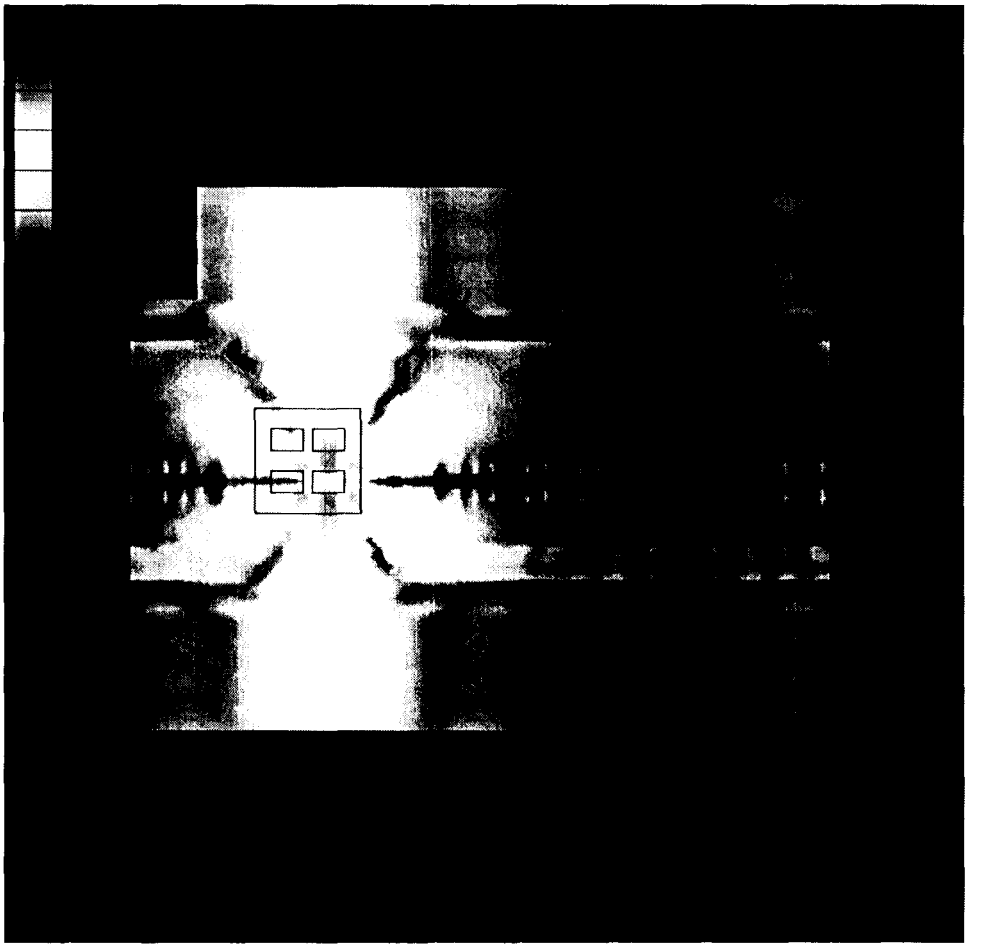


Figure 20. Phi component of magbetic field on surface of revoultion boundary for conformal patch antenna.

V1. SUMMARY AND FUTURE CHALLENGES

This has chapter described the motivation, formulation and implementation of a finite element method for the calculation of electromagentic fields using massively parallel processors. Key points of this work include a) an efficient implementation of a surface of revolution integral equation for coupling the exterior radiation boundary condition to the computational mesh, b) the use of a matrix decomposition onto the processors that differs from mesh decomposition strategies, c) the development of a parallel quasi-minimum residual iterative algorithm for distributed memory massively parallel computers, and d) the extension of the parallel scattering code to antenna modeling. From this experience, two areas stand out as future challenges—mesh generation and more efficient sparse matrix equation solvers.

The mesh generation stage involves creating the computer description of the geometry and a mesh of the region in and about the scatterer or antenna. For target sizes that require millions of elements, this is a daunting process. The mesh is generated in the region containing penetrable materials and out to the minimal surface of revolution surrounding the target or antenna. Creating elements of relatively uniform shape that conform to the targets geometry and have the necessary density is a difficult task. Additionally, it is

difficult to assess the quality a mesh after it has been generated with millions of elements. One possible approach for improving this situation is to apply adaptive mesh refinement strategies that allow a very coarse initial mesh to be generated, with refinement being automatically performed within the finite element code [17].

The second challenge involves creating sparse matrix solution algorithms that are efficient for many thousands (or more) of right-hand-sides. If an iterative solver is used, convergence should be relatively uniform for all right-hand-sides, and on a parallel machine the solver should maintain data load balance as well as minimal communication as the problem size grows. The solver outlined in this chapter uses a matrix decomposition by row-slab partitioning following reordering that produces data structures that generally allow a balanced matrix-vector multiplication in the iterative solver. The data load balance was almost exactly uniform, while the communication overhead was moderately small and similarly uniformly balanced over the machine for the majority of problems considered. For scaled-sized problems, the communication time was at most 15% of the total matrix-vector multiplication time. Even bringing this percentage down to zero would not lead to a major improvement in the overall performance of the code. However, major improvements are possible in two areas: the local multiplication and the number of quasi-minimum residual iterations.

First, the performance of the local portion of the sparse matrix-dense vector multiplication could be improved. This is dependent on the sparse data-

storage structure of the matrix and how it is loaded into the local cache. The relative sparsity of the reordered row slab of the matrix causes the multiplication to jump around in the cache as it loads the elements of the X vector. If these local row slabs were reordered in such a way as to obtain a more dense matrix, the local performance could increase dramatically.

Second, an efficient parallel preconditioned, or block iterative solver could decrease the number of iterations needed in the matrix equation solution. Naturally, the preconditioned must not increase either the overhead in setting up the problem or obtaining the final solution more than it saves by lowering the iteration count. The block solver also must not increase the time per iteration more than the amount it saves by lowering the iteration count.

ACKNOWLEDGMENT

The authors wish to gratefully acknowledge the support of Jean Patterson, manager of the task *Research in Parallel Computational Electromagnetics*, Vahraz Jamnejad, a member of this task at JPL, and Mike Heroux of Cray Research, who assisted in developing an understanding of various sparse matrix-dense vector multiplication formulations. The JPL/Caltech Supercomputer used in this investigation was provided by the NASA Offices of Mission to Planet Earth, Aeronautics, and Space Science. The Cray T90 used in this investigation was provided by the Information Services Department of Cray Research. The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of

Technology, under contract to the National Aeronautics and Space Administration (NASA).

REFERENCES

[1] T. Cwik, C. Zuffada, and V. Jamnejad, "The Coupling of Finite Element and Integral Equation Representations for Efficient Three Dimensional Modeling of Electromagnetic Scattering and Radiation," *Finite Element Software for Microwave Engineering,* T. Itoh, G. Pelosi, P. Silvester, Editors, John Wiley and Sons, Inc., New York, 1996.

[2] C. Zuffada, T. Cwik, and V. Jamnejad, "Modeling Radiation with an Hybrid Finite Element–Integral Equation–Waveguide Mode Matching Technique," *IEEE Trans. Antennas Propag.*, Vol. 45, No. 1, pp.34-39, Jan. 1997.

[3] T. Cwik, C. Zuffada, and V. Jarnnejad, "Modeling Three-Dimensional Scatterers Using a Coupled Finite Element-Integral Equation Representation," *IEEE Trans. Antennas Propag.,* vol. 44, no. 4, pp. 453–459, April 1996.

[4] Lee, J. and Mittra, R. (1992), "A Note On The Application Of Edge-Elements For Modeling 3-Dimensional Inhomogeneously-Filled Cavities," *IEEE Transactions On Microwave Theory and Techniques, vol. 40, no. 9,* pp. 1767–1773, Sept.

[5] Medgeysi-Mitschang, L., and Putnam, J. (1984), "Electromagnetic Scattering from Axially Inhomogeneous Bodies of Revolution," *IEEE Transactions on Antennas and Propagation, vol. AP-32,* pp. *797–806.*

[6] B. Nour-Omid, A Raefsky, and G. Lyzenga, "Solving Finite Element Equations on Concurrent Computers," American Sot. *Mech. Eng.,* A Noor Editor, pp. 291-307, 1986.

[7] A. Pothen, H. Simon and K. Lieu, "Partitioning Sparse Matrices with Eigenvectors of Graphs," SIAM J. Matrix Anal. Appl., vol. 11, pp. 430–452, 1990.

[8] B. Hendrickson and R. Leland, "An Improved Spectral Graph Partitioning Algorithm for Mapping Parallel Computations, " *S/AM J. Sci. Comput.,* vol. 16, pp. 452-469, 1995.

[9] G. Karypis and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *Technics/ Report TR 95-035,* Department of Computer Science, University of Minnesota, 1995.

[10] T. Cwik, D, Katz, C. Zuffada, and V. Jamnejad, "The Application Of Scalable Distributed Memory Computers To The Finite Element Modeling Of Electromagnetic Scattering," *Intl. Jrnl. Numer. Methods. Eng.,* accepted for publication.

[11] R. Freund, "Conjugate Gradient-Type Methods for Linear Systems with Complex Symmetric Coefficient Matrices," *S/AM J. Stat. Comput,* vol. 13, no. 1, pp. 425-448, Jan, 1992.

[12] A. George and J. Liu, "Computer Solution of Large Sparse Positive Definite Systems," Prentice Hall, New Jersey, 1981.

[13] J. Lewis, "Implementation Of The Gibbs-Poole-Stockmeyer And Gibbs-King Algorithms," *ACM Trans. On Math. Software,* vol. **8,** pp. 180-189, *1982.*

[14] T. Cwik, R. van de Geijn, and J. Patterson, "Application of Massively Parallel Computation to Integral Equation Models of Electromagnetic Scattering (Invited Paper)," *J. Opt. Sot. Am. A, vol. 11,* **no.** *4,* pp. 1538-*1545,* April 1994.

[15] Douglas J. Taylor, "Scattering and Extinction by Anisotropic Spheres," Journal of Wave-Material Interaction Vol. 3, no. 4, Oct. 1988, pp 327-339.

[16] Larry W. Epp, Daniel J. Hoppe, Gilbert C Chinn, "Scattering from ferrite bodies of revolution using a hybrid approach," IEEE Microwave and Guided Wave Letters, **Vol. 5, No. 4, Apr. 1995, ppl 11-1 13; and personal communications with authors.**

[17] T. Cwik and J. Lou, "Error Estimation and h-Adaptivity for Optimal Finite Element Analysis", 1997 *Digest, USNC/URSI Radio Science Meeting, APS Digest,* **p.** *664-667,* **Montreal, CA July 13-18, 1997.**